

**Los Portales**  
**An Online Ticket Buying System**  
**Final Report**

Justyn Rippie

Al Yazzie

Matt Mitchell

Nick Thompson

## Contents

1.	Customer Requirements.....	3
2.	System Requirements .....	4
3.	Cost Estimation.....	4
4.	Project Execution.....	6
5.	Overall Software Description .....	8
6.	External Interface Requirements.....	22
7.	System Features .....	22
8.	Non-Functional Requirements.....	38
9.	Software Testing Plan.....	39
10.	Software Testing Results .....	40
11.	User Documentation .....	44
12.	Future of the Software .....	51

# **1. Customer Requirements**

## **1.1 Background Information**

Theater Los Portales is seeking an online booking system to allow customers to book a ticket for any play. Los Portales currently doesn't have a means for customers to purchase tickets before any show. A potential customer must be able to create a new account which will allow them to interact with the system. Customers must be able to log into their account to purchase ticket(s) using either a credit or debit card. Tickets for any show must be presented in a graphical interface that shows each seat and row in the theater. The customer can then click a seat(s) of their choice. Once the user has selected the seat, tickets should go into a shopping cart and be ready for a user to checkout. Likewise, a customer should be able to edit their shopping cart for any changes. Once the customer is checked out and is notified their form of payment is accepted, an email will be sent to the customer with their purchase details.

Moreover, Los Portales requires that an administrator be able to log into the system to assign prices to each seat for a specific play. The administrator must also be able to run a report for each play to see how many tickets have been sold and the total cost sold for each play.

## **1.2 Current Environment**

Currently the Los Portales theater does not have the means for customers to purchase tickets through an online system. The current method of sales requires a customer to come to the theater to pay for the ticket with cash currency. The theater admin uses a pen and paper to keep track of ticket sales for the play that is showing and then enters the data enter a computer.

## **1.3 Specific Requirements**

The following are specific requirements that must be provided to the client:

- I. A graphical system to select seats. Seats sold must be marked in red and seats available must be marked in green.
- II. The system must display all plays and their show times and dates within a year
- III. A customer must be able to select multiple seats from one or many shows
- IV. A customer must be able to add a seat(s) to the shopping cart.
- V. The system needs to have a customer registered in order to buy seats
- VI. The shopping cart should be editable for the customer.
- VII. The system must inform the user of a successful transaction showing the seats specific to a play
- VIII. The system must always be online
- IX. A theater admin can create, update and delete seat prices for a specific play
- X. A theater admin must be able to generate reports of sales for a specific play

## 2. System Requirements

The MVC app will need to be published to a Web Server and will need to have a database created on a server or the computer that this website will run from. This new database connection will then need to be updated in the Connections.

## 3. Cost Estimation

### 3.1 General System Characteristics

The list below provides 14 General System Characteristics that any piece of software may use. Each category is rated from 0-5. Below is a table of the 14 characteristics with the rank. Also provided is the table to explain each software characteristic and ranking.

General System Characteristic	Rank
Data Communications	3
Data Processing	4
Performance	0
Heavily Used Configuration	0
Transaction Rate	0
Online Data Entry	0
End-User Efficiency	2
Online Update: 0	0
Complex Processing: 0	0
Reusability	1
Installation Ease	0
Operational Ease	0
Multiple Sites	0
Facilitate Change	0

### 3.2 Unadjusted Function Point (UFP) Calculation

The UFP provides a basic calculation of the systems requirements that the user will use. Essentially it is a way to score the system based on the needs of user and how the system will interact with the user.

Function Type	Count		Complexity	Total
External Inputs	12	x	3	36
External Outputs	3	x	5	15
External Enquired	1	x	4	4
Internal Logic Files	1	x	7	7
External Interface Files	1	x	10	10
Count Total				72

### 3.3 Value Adjustment Factor Calculation

The Value Adjustment Factor (VAF) is based on the 14 General System's Characteristics in Section One.

$$\text{The equation is: } VAF = 0.65 + \left[ \frac{\sum_{i=1}^{14} C_i}{100} \right].$$

$$VAF = 0.65 + (0.01 \times 10) = 0.75$$

### 3.4 Adjusted Functional Points Calculation

The calculation for the adjusted functional points (AFP) follows the equation:  $AFP = UFP \times VAF$ .

$$\text{Calculation: } AFP = 72 \times 0.75 = 54$$

**Lines of Code Calculation** It is important we estimate how many lines of code the software developers will need to code. This excludes any pre-generated code from an Integrated Development Environment (IDE) and any Application Programming Interface (API). The equation is:  $LOC = AFP \times SLOC$ , where the SLOW is derived from the function point languages table provided here: <https://www.qsm.com/resources/function-point-languages-table>

$$LOC = 54 \times 54 = 2916$$

### 3.5 Additional Costs:

Additional costs can cover infrastructure, training, equipment, etc. The client will need the following:

- A. Cloud Hosting Provider (includes https) \$49 month
- B. Training = 6750

### 3.6 Total Cost Calculation

The total cost calculation is based on the Constructive Cost Model (COCOMO). It is an algorithmic model, and the cost is estimated by the lines of code the programmers will code. The model of this calculation follows a basic model where we only use lines of code to estimate the cost. The type of project is an organic project which entails less than 50k lines of code, a small project, a project that is specific to the client. Below is the cost model with the final estimated cost.

#### Basic Model:

$$\text{Effort} = E = a(KLOC)b \text{ (person/month)}$$

$$\text{Time} = T = c(E)d \text{ (months)}$$

$$\text{Average Staff Size} = P = E/T \text{ (persons)}$$

$$\text{Productivity} = Pr = LOC / E \text{ (LOC/person_month)}$$

$$\text{Cost} = C = (E \times \$\$ \text{ person month}) + \text{other costs}$$

Software Project	a	b	c	d
Organic	2.4	1.05	2.5	0.38

Lines of code rounded: 2.9 KLOC

$$\text{Effort} = E = 2.4(2.9)^{1.05} = \text{approx. } 7.3 \text{ person/month}$$

$$\text{Time} = T = 2.5(7.3)^{0.38} = 5.3 \text{ months}$$

$$\text{Average Staff Size} = P = 7.3/5.3 = 1.4 = 1 \text{ person}$$

$$\text{Productivity} = Pr = 2916/4.4 = 662.7 \text{ LOC/month}$$

$$\text{Cost} = C = (7.3 * 6750) + 6799 = \$56,074$$

## 4. Project Execution

### 4.1 Period of Performance

The project for Los Portales will be divided into four stages. The first stage for Planning, the second stage for Project Development, the third stage for Development and Testing. During the second and third stages software testing will be continually conducted. Lastly, the fourth stage will finalize the product for release to the client.

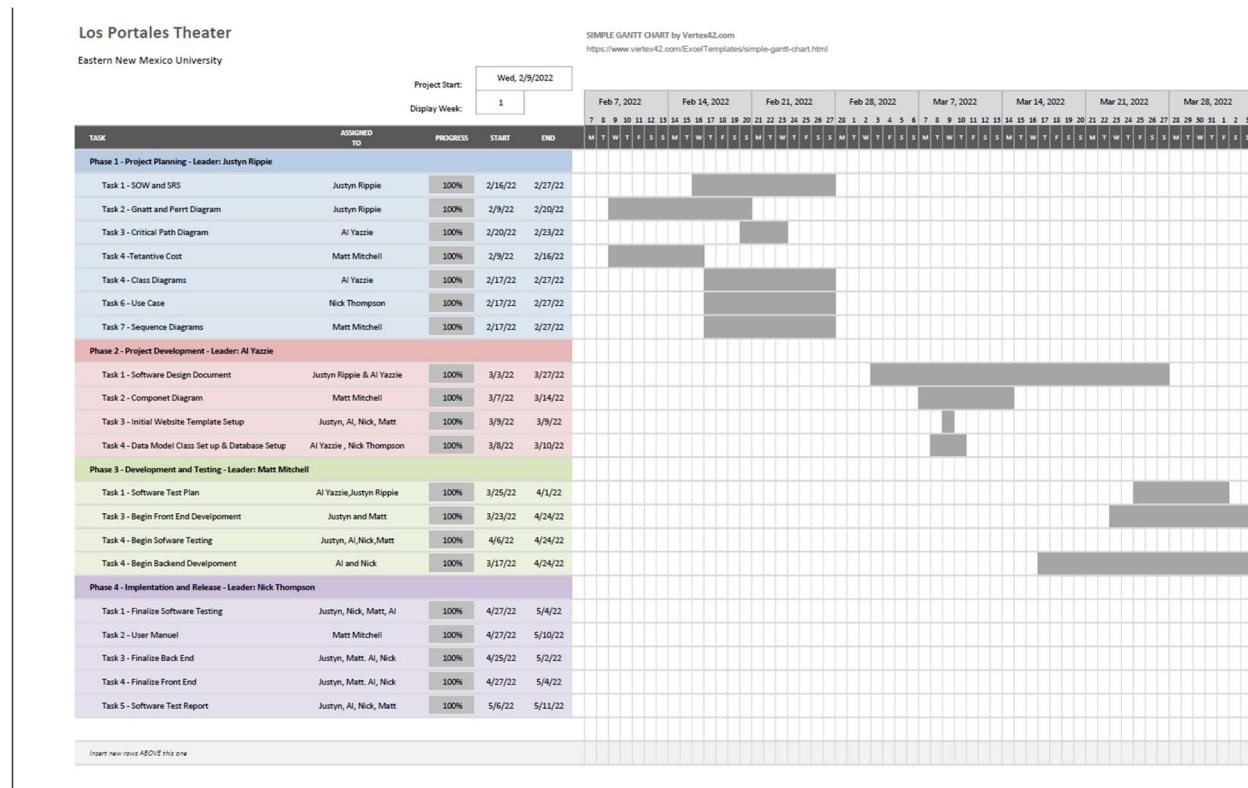
The client will have a full functioning system delivered by May 11, 2022.

### 4.2 Milestones

Milestone	Estimated Delivery
<b>Phase 1: Project Planning</b>	2/27/2022
<b>Phase 2: Project Development</b>	3/27/2022
<b>Phase 3: Development and Testing</b>	4/17/2022
<b>Phase 4: Implementation and Release</b>	5/11/2022

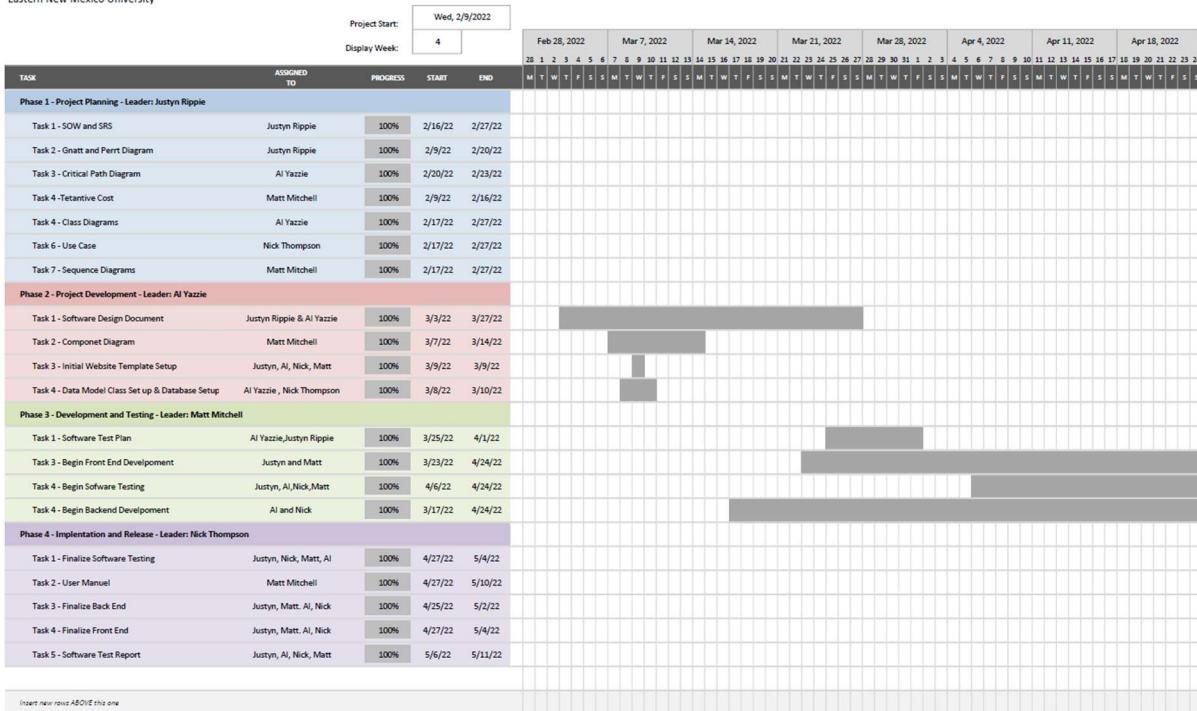
### 4.3 Gantt Chart Progress

Below are images showing progress of the milestones in 4.2.



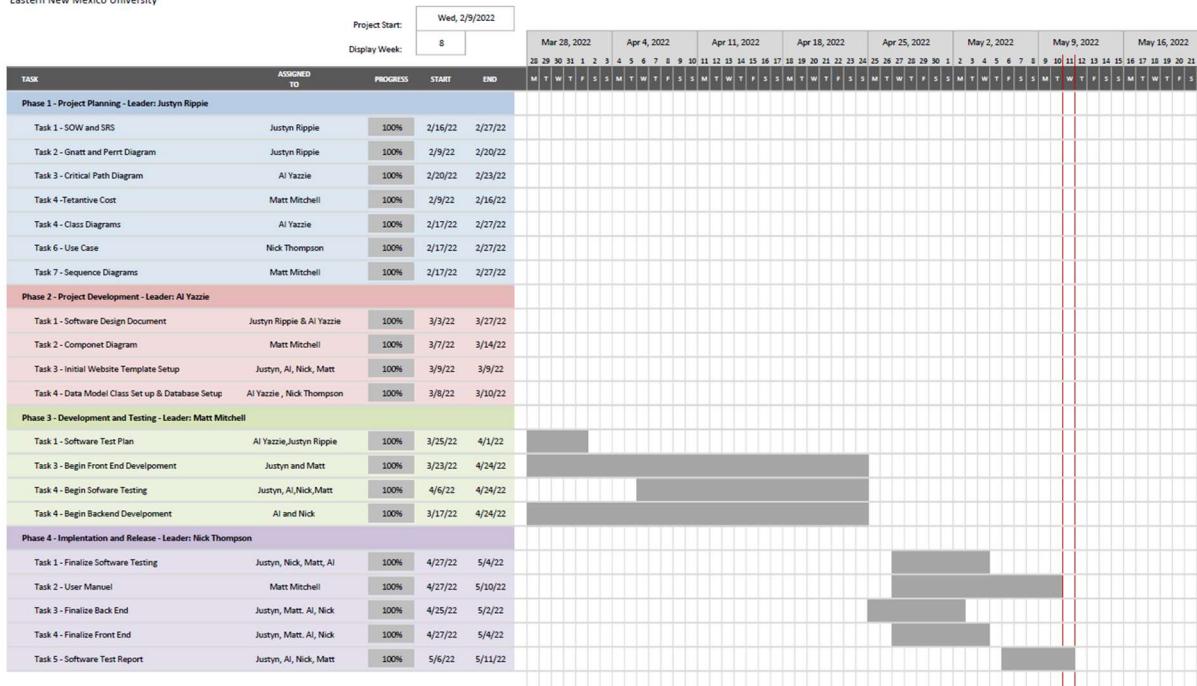
## Los Portales Theater

Eastern New Mexico University



## Los Portales Theater

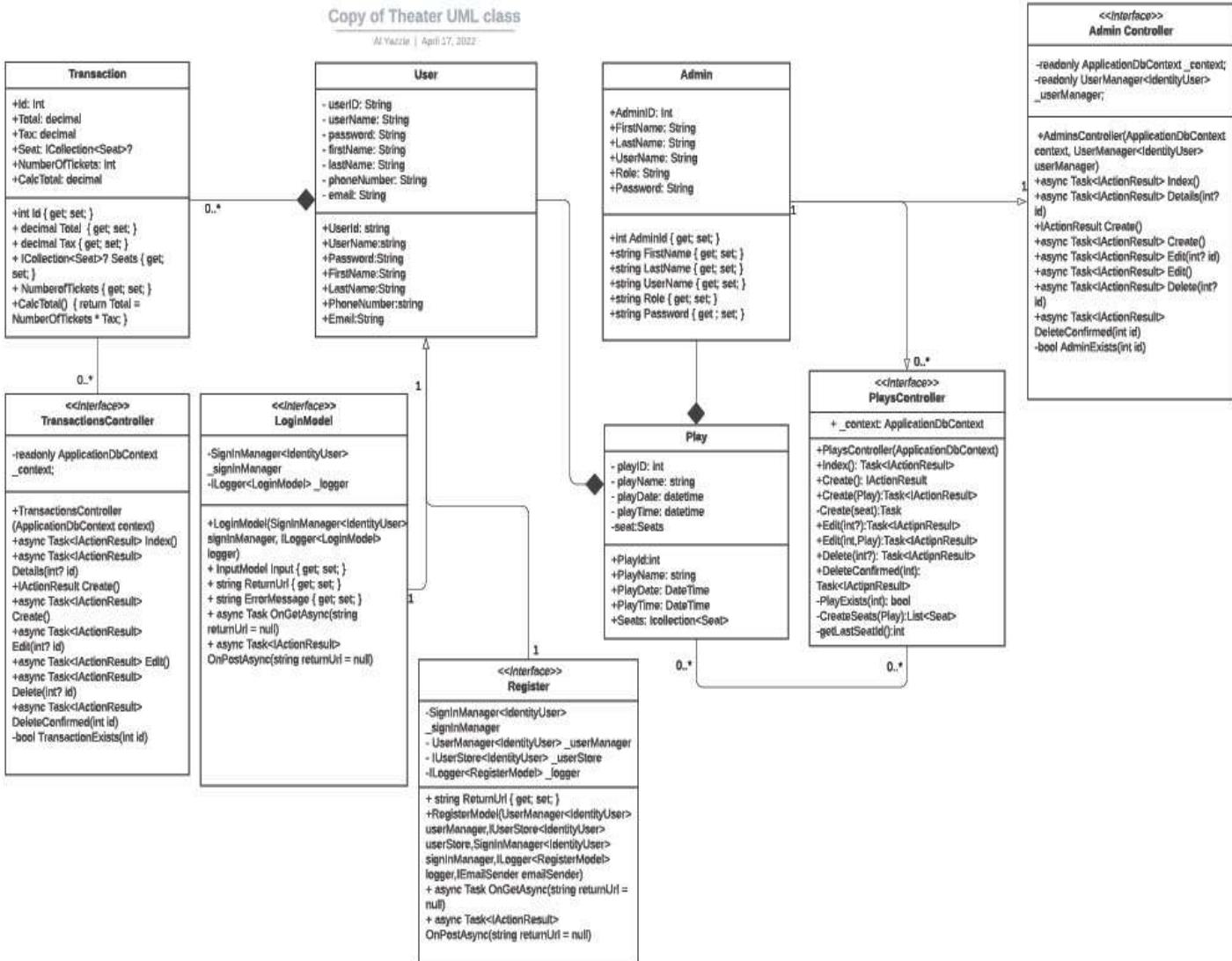
Eastern New Mexico University



## 5. Overall Software Description

### 5.1 User Classes and Characteristics

Below is an image of the Class UML Diagram



### 5.2 Operating Environment

The ticketing and reporting web application will be hosted by a cloud hosting provider who will assume responsibilities for the following services:

- Microsoft Sever SQL Database
- HTTPS Certificate
- Capable of hosting Microsoft .NET-Core Web Application
- Domain Name Service

## 5.3 User Documentation

A User Manual will be provided upon software delivery. The following topics will be included in such document:

- Creating a show
- Updating a show
- Canceling a show
- Generating a sales report
- Logging into the system
- Creating a new Admin account

For Customer inquiries, an FAQ will be available on the website.

## 5.4 Assumptions and Dependencies

Assumptions will help the vendor define timelines, the scope of the project, and effort needed to complete the project.

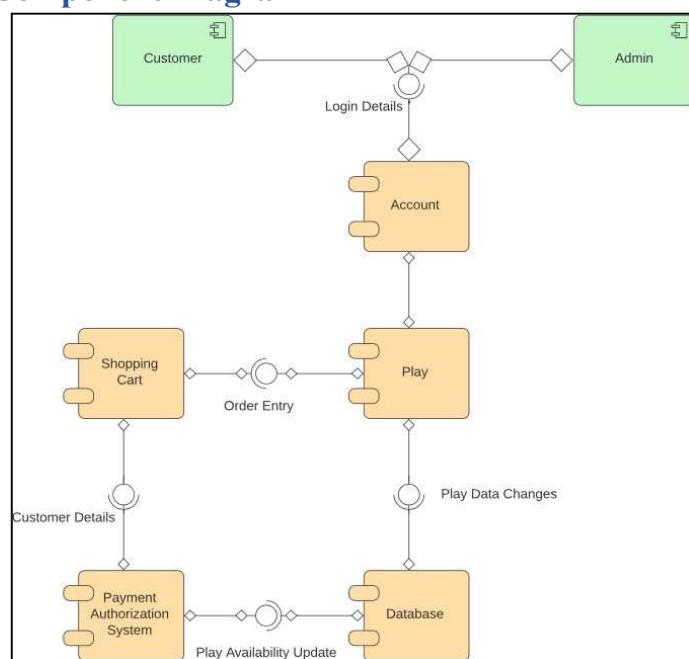
### Technology Assumptions

- The client has no current Information Technology Infrastructure
- The client has one desktop computer that is connected to the internet
- The client will need a cloud hosting platform to host the system and will include:
  - Microsoft Sever SQL Database
  - HTTPS Certificate
  - Capable of hosting Microsoft .NET-Core Web Application
  - Domain Name Service

## 6. Software Design

### 6.1 Architectural and component-level design

#### 6.1.2 Component Diagram



## 6.2 Play Class

The Play Class represents the Los Portales theater play details. The class is comprised of several data points including a unique identifier for each Play, the Play Name, Play Date, Play Time, when the admin created the record and if/when a play is modified.

### 6.2.1 Processing narrative

When the Admin adds a new play to the system, a Play object is created. This object is responsible for storing information unique to the Play. This includes the following:

- PlayID
- PlayName
- PlayDate
- PlayTime

Any time information related to a Play is required, the Play object is called. For example, the Admin needs to create or modify Play information, each are identified using its unique PlayID. When a new Play is created/modified, the current date and time are captured and written to the database to the CreateDateTime and/or ModifiedDateTime column for tracking changes to the database.

### 6.2.2 Processing detail

The only methods in the class are accessors, mutators, and a method to interact with the database.

### 6.2.3 Design Class hierarchy

The Play class has no parent or child classes.

### 6.2.4 Restrictions/limitations

There are no restrictions.

### 6.2.5 Performance issues

At any one time, there is only one Play object created or modified by the admin therefore no performance issues are present. The only issue that could arise would occur if the database server is being serviced thus, temporarily halting incoming and outgoing connections until the server is operational again.

### 6.2.6 Design constraints

The major constraint for this class is each new Play object will need to include a Play Name, Play Date and a Play Time. The table on the database does not allow null values in these fields.

### 6.2.7 Processing detail for each operation

Accessors/mutators

- These methods are used to get and set Play object information as needed. All fields can be modified except the PlayID which is a primary key on the table.

## 6.3 User Class

The User Class represents the Los Portales customers. The class is comprised of several data points including a unique identifier for each User, a Username, Password, First Name, Last Name, Phone Number and Email address when a User object is created or when a User object is modified. This class is utilizing built in MVC Identity functionality for the creation of Roles within the application

### 6.3.1 Processing narrative

When a new Account is created a new User object is created. This object is responsible for storing information unique to each User. This includes the following:

- UserID
- UserName

- Password
- FirstName
- LastName
- PhoneNumber
- Email

Any time information related to a User is required, the User object is called and identified using its unique UserID. When a new Account is created/modified, the currentdate and time are captured and written to the database to the CreateDateTime and/or ModifiedDateTime column for tracking changes to the database.

### **6.3.2 Processing detail**

Several methods within this class are accessors or mutators.

#### **6.3.2.1 Design Class hierarchy**

The User class has no parent or child classes.

#### **6.3.2.2 Restrictions/limitations**

There are no restrictions.

#### **6.3.2.3 Performance issues**

Several User accounts may be created from several sources which could potentially decrease available resources, nonetheless the UserID primary key on the table would prevent duplicate accounts. Like the Play class, the only issue that could arise would occur if the database server is being serviced thus, temporarily halting incoming and outgoing connections until the server is operational again.

#### **6.3.2.4 Design constraints**

The major constraint for this class is each new User object will need to include a Username and a Password. The table on the database does not allow null values in these fields.

#### **6.3.2.5 Processing detail for each operation**

Accessors/mutators

- o These methods are used to get and set User object information as needed. All fields can be modified except the UserID which is a primary key on the table.

## **6.4 Transaction Class**

The Transaction Class represents the Shopping Cart and a Finalized processed payment within the Los Portales webpage. The class is comprised of several data points including a unique identifier for each User, the Number of Tickets, Selected Seats, the Payment Method, Tax, and a Total when a User purchases a ticket.

### **6.4.1 Processing narrative**

When a purchase is about to occur, this object is responsible for storing information unique to each Transaction. This includes the following:

- CustomerID
- Total
- Tax
- SelectedSeats
- NumberOfTickets
- PaymentMethod

Before a Transaction is completed, each Transaction is labeled with a unique TransactionID.

### **6.4.2 Processing detail**

There are two accessor methods which work with Calculating the Total Sale. The last is a method to interact with the database.

#### **6.4.2.1 Design Class hierarchy**

The Play class has no parent or child classes.

#### **6.4.2.2 Restrictions/limitations**

There are no restrictions.

#### **6.4.2.3 Performance issues**

Several Transactions can be performed from several sources which could potentially decrease available resources, nonetheless the TransactionID primary key on the table would still separate each transaction with their own identifier. Like the Play class, the only issue that could arise would occur if the database server is being serviced thus, temporarily halting incoming and outgoing connections until the server is operational again.

#### **6.4.2.4 Design constraints**

The major constraint for this class is each new Transaction will need to include the PlayID, the UserID and the Total. The table on the database does not allow null values in these fields. This is also to link which user purchased each ticket.

#### **6.4.2.5 Processing detail for each operation**

Accessors/mutators

- These methods are used to get Play information as needed. All fields can be modified except the TransactionID which is a primary key on the table.

### **6.5 Admin Class**

The Admin Class represents the Los Portales customers and the theater admin. The class is comprised of several data points including a unique identifier for each Admin, a Username, Password, First Name and Last Name and when an Admin object is created or when an Admin object is modified. This class is utilizing built in MVC Identity functionality for the creation of Roles within the application.

#### **6.5.1 Processing narrative**

When a new Admin is created a new Admin object is created. This object is responsible for storing information unique to each User. This includes the following:

- AdminID
- UserName
- Password
- FirstName
- LastName

Any time information related to a User is required, the User object is called and identified using its unique UserID. When a new Account is created/modified, the current date and time are captured and written to the database to the CreateDateTime and/or ModifiedDateTime column for tracking changes to the database.

#### **6.5.2 Processing detail**

Several methods within this class are accessors or mutators. There is one default constructor method in this class.

#### **6.5.2.1 Design Class hierarchy**

The Admin class has no parent or child classes.

#### **6.5.2.2 Restrictions/limitations**

There are no restrictions.

### **6.5.2.3 Performance issues**

Admin account creations are uncommon, unlike User accounts, nonetheless the AdminID primary key on the table would prevent duplicate accounts. Like the Play class, the only issue that could arise would occur if the database server is being serviced thus, temporarily halting incoming and outgoing connections until the server is operational again.

### **6.5.2.4 Design constraints**

The major constraint for this class is each new Admin object will need to include a Username and a Password. The table on the database does not allow null values in these fields.

### **6.5.2.5 Processing detail for each operation**

Accessors/mutators

- o These methods are used to get and set Admin object information as needed. All fields can be modified except the AdminID which is a primary key on the table.

## **6.6 User interface design**

A user interface is crucial to this software as it will help make the process of purchasing a ticket much easier for the customer. As of right now we have identified 8 user interfaces that are critical to ensure the task of purchasing a ticket and ensuring the theater admin can complete their job.

### **6.6.1 Description of the user interface with Objects and Actions**

The user interface will display the following:

1. Log in page for the user
  - a. Username
  - b. Password
2. Register page where user can create an account
  - a. Password
  - b. Name
  - c. Date of birth
  - d. Email address
3. A list of plays for the user which they can click on to view the seating chart
4. Seating chart for each play where each seat has the price listed and an available seat is green and taken seat is red.
5. A check out page
6. A shopping cart for tickets must be editable and the user can delete items
7. A sales report page for the theater admin can select a play from a drop-down menu. The interface will also ask the admin which fields the admin wants to know about in a check box: seats sold, seats available, total revenue.
8. A page for plays in which the theater admin can Create, Delete, Update plays.
  - a. Create page – Admin can enter in the name, date, and time of play
  - b. Delete page - Admin can delete the play
  - c. Update Play - Admin can update the name, date, time, and seating prices

## 6.6.2 Screen images - Los Portales Theater GUI

CHECKOUT

Credit Card Number

Name on Card

CCV

Expiration Date

**PAY**  Save Card

Shopping Cart

Play	Number of Seats	Cost	Action	
Play 1	4	\$120	Delete	Edit
Play 2	10	\$500	Delete	Edit
Play 3	2	\$200	Delete	Edit
Play 4	4	\$300	Delete	Edit

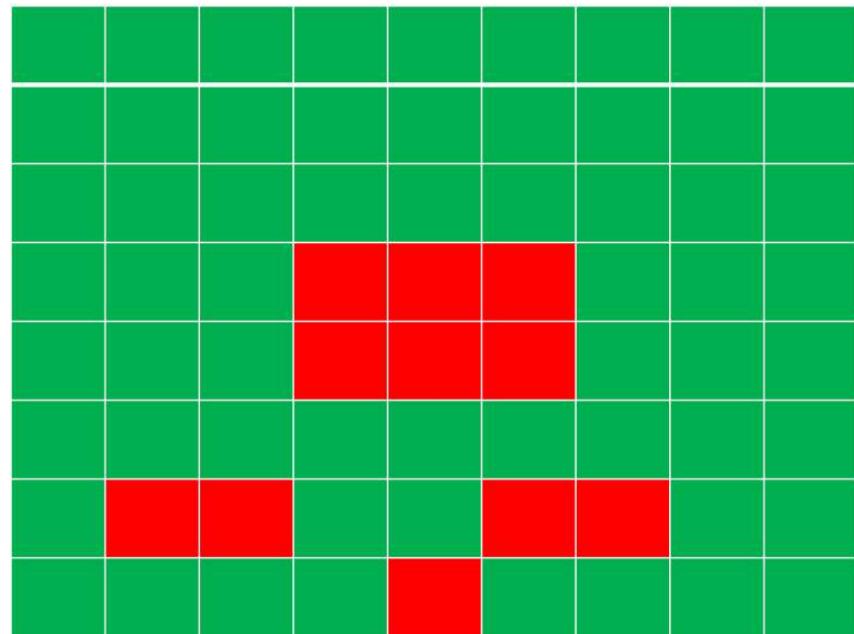
**Total Cost** **\$1,120**

**Checkout**

### Available Plays

Play Name	Date	Time	Action
Play 1	12/02/2022	7:00 PM	Purchase Tickets
Play 2	06/05/2022	1:00 PM	Purchase Tickets

### Seating Chart for <Play Name>



### Log Into Your Account

UserName

Password

**Login**

### Create an Account

An Account is required to purchase a ticket for a play

Name

Address

Email Address

Password

Confirm Password

**Create Account**

Administration

**Create New Play**

Play Name	Action		
Play 1	Update	Delete	Assign Ticket Prices
Play 2	Update	Delete	Assign Ticket Prices

**Sales Report**

Name of Play Drop Down	
<input type="checkbox"/>	Seats Sold
<input type="checkbox"/>	Seats Available
<input type="checkbox"/>	Total Revenue

**Create a Play**

Name	<input type="text"/>
Date	<input type="text" value="12/01/2999"/>
Time	<input type="text" value="12:00 PM"/>
<b>Create</b>	

### 6.6.3 Interface design rules

The user interface shall follow the rules below:

- Consistency
- Use of shortcuts
- Provide feedback to the user
- Design dialog to yield closure
- Simplify to prevent errors
- Easily reverse actions
- Give users control
- Reduce short-term memory load

#### **6.6.4 Components available**

Listed below are GUI component that will be implemented in HTML.

1. Textbox – allows a user to input data to certain data that is being asked from them.
2. Label – Helps label certain elements on the GUI to guide users.
3. List Box – Displays a list of items in which the user can select an option from.
4. Table – Ensures data is presented neatly
5. Checkbox – Allows the user to check a box when input is needed to perform an operation
6. Button – Allows a user to click and then performs an operation after inputting data to a checkbox, textbox, or list box
7. Link – Has a hyperlink that can help user easily navigate to certain areas of the app by embedding a URL

### **6.7 Restrictions, limitations, and constraints Time**

Due to the project being a 15-week semester for a semester of school, the team has very limited time capacity to build, test and implement a working software to the client. The development phase needs to be extremely monitored as the team should not waste time during this phase to build a working software.

#### **6.7.1 Testing Issues**

To ensure the software executes properly, we need to incorporate testing into our development phase to ensure the software is bug free. The testing should ensure proper input is taken and the proper output is outputted.

#### **6.7.2 Classes of tests**

Here, we describe how we will test the software as it is being developed.

##### **Unit Testing**

We will use white box testing and write unit test to ensure each component is functioning correctly. This will be important for the components that interact with the database.

Ensuring that data is written correctly to the database is critical to ensure the app functions how the client wants expects it to respond.

Likewise, the development team will write unit cases for actions of registering a user, creating a play, updating a play, removing a play, and completing a transaction. This data is stored in the database, and we must ensure that each action performs as expected.

##### **Component Testing**

With many different components within this software, it is critical we test each component as well and ensure it functions correctly. For example, we shall test the graphical user interface within an action that requires a database action. We can perform a unit test for this and generate a script to test the GUI to ensure that it only accepts the correct format for input. Next, we shall validate the information to whatever backend function that needs to execute.

##### **Use-Case testing**

We should also test our software by examining our use cases. We should ensure that the use case and the results listed out should match how the system is interacting. This type of testing can ensure the front-end and back-end of the

application are communicating properly and the expect results are seen.

### User Testing

When the software is functioning as expected and all the above testing is complete, we shall consult the client to test the software. Having the user test the software before releasing it will help guide the team into what the client needs, and the requirements are met. Having a clean and user-friendly interface will help the user navigate the software and the client should have final say in how they want the software to look so that they may use the software without distractions.

### 6.7.3 Expected software response

There should be expected results from the software while testing. While testing actions that connect to the database, we should expect these methods will write when a write action is performed, a read action indeed reads from the database, and a delete method deletes the proper data from the database. There should also be ways the software ensures a failed action doesn't edit the database and informs a user that an action failed and how they should correct it.

### 6.7.4 Identification of critical components

Listed below are critical components of the software:

#### 6.7.4.1 Transaction

The transaction component is the backbone of this application. During the test phase this component needs to be tested to ensure it functions properly. That is, we need to make sure a successful transaction is successful, and that a failed transaction doesn't purchase the tickets and does not pull money from a user's card.

#### 6.7.4.2 Play Creation

Ensure the play creation component of the administrator side of the application is critical to the application. If the theater admin cannot create a play, the user cannot purchase tickets for a play. Likewise, this also entails the theater admin can assign prices to the seats so that the user may see them to purchase.

#### 6.7.4.3 Account Creation

Creating an account is another critical component to the software. If a customer cannot create an account, they will have no way to purchase tickets. This module should ensure that a user will always have a way to create an account and give proper feedback if an account cannot be generated and refer the user on how to set up an account.

## 6.8 Data Design

The Los Portales Theater is a C# based Model View Controller (MVC) web application divided into two parts: client-side and server-side.

### 6.8.1 Internal Software Data Structure

On the client side, user interaction with the data is managed by various Views presented via web forms. The client side will be implemented using a combination of HyperText Markup Language (HTML) and JavaScript. Data is organized in various tables stored in a MySQL database, a breakdown of the various tables and columns are described later within this Design Section (*Database description*). The data will be retrieved from the database via HTTP Get requests from the Server upon webpage initialization.

The data structure on the server will be implemented using C#. Data to be sent to the MySQL database is exchanged with embedded SQL statements (i.e., Merge, Delete, Select, etc.).

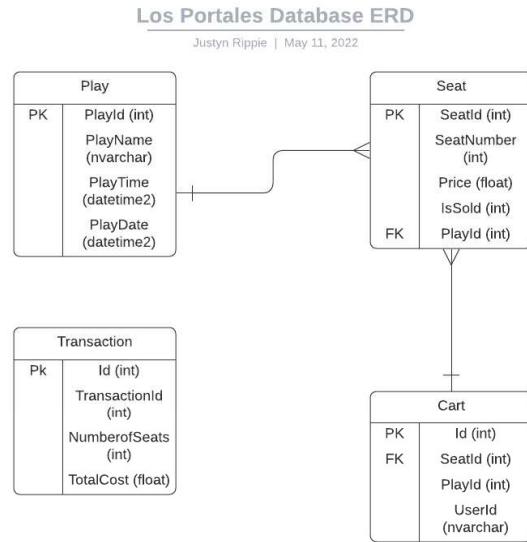
### **6.8.1 Global Data Structure**

The key data components for this web application revolve around a centralized database. The webpage will request data from the server via various HTTP Get and Post methods.

### **6.8.2 Temporary Data Structure**

Temporary data structures in the Los Portales Theater webpage refer to the various SQL queries and scripts which will only be run based on the source transaction.

### 6.8.3 Entity Relational Diagram of the Database



## **7. External Interface Requirements**

### **7.1 User Interfaces**

The software will contain a user interface so that the user can interact with the system. This user interface will be Graphical Interface designed with HTML, JavaScript and ASP.NET-Core.

### **7.2 Software & Communication Interfaces**

The client will need a cloud hosting platform to host the system and will include:

- Microsoft Sever SQL Database
- HTTPS Certificate
- Capable of hosting Microsoft .NET-Core Web Application
- Domain Name Service

The cloud hosting provider requires the following communication standards:

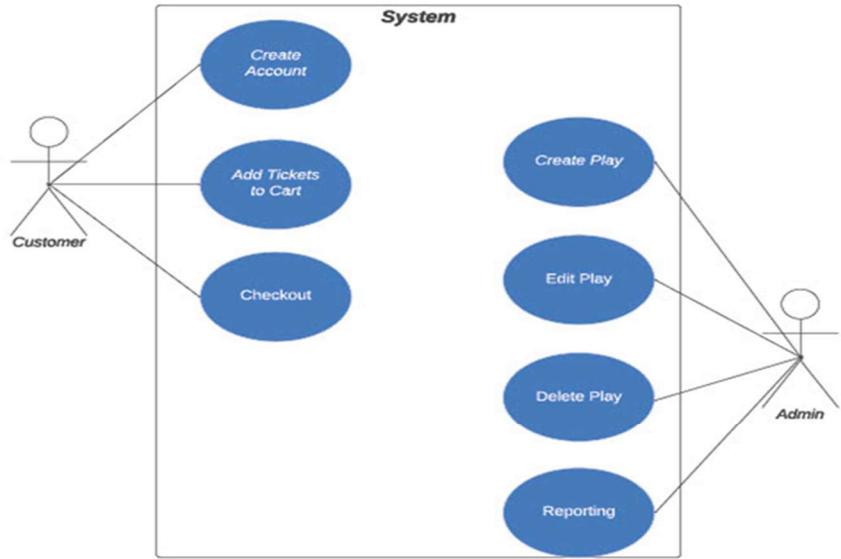
- HTTPS (Port 80, 443)
- SMTP Server (Port 25)
- DNS (Port 53)
- NTP (Port 123)

## **8. System Features**

### **8.1 Use Cases and Diagram**

Below are tabular listed use cases for user and system interaction. Included is also a diagram on the interaction of the user with the specified system requirements.

Use case diagram



Use Case ID:	001		
Use-Case Name:	Customer Registers		
Created By:	Justyn Rippie	Last Updated By:	
Date Created:	2/23/2022	Date Last Updated:	
Actors:	Customers		
Description:	<p>The customer will need to register with the booking system by providing their: first name, last name, email address, phone number, date of birth.</p> <p>By registering with the booking system, this allows them to complete a purchase and allow an administration to run reports on ticket sales.</p>		
Preconditions:	None		
Postconditions:	User will be able to select tickets and purchase tickets.		
Priority:	High		
Frequency of use:	Low		
Normal Course of Events:	A new customer will visit the booking systems and will require them to create an account in order purchase tickets.		
Alternative Courses:	If a customer tries to create a preexisting account, the system will notify the user of their account and ask them to log in to complete the purchase.		
Exceptions:	None		
Includes:	None		
Special Requirement	None		
Assumptions:			
Notes and Issues:	N/A		

Use Case ID:	002		
Use-Case Name:	Customer adds ticket to shopping cart		
Created By:	Justyn Rippie	Last Updated By:	
Date Created:	2/23/2022	Date Last Updated:	
Actors:	Customers		
Description:	A customer will be presented with a list of plays with and the dates they are showing. They will then click on the name of the play to select tickets and be presented with the theater seats to choose a seat(s). When they click on a seat(s) they will be asked if they want to add this to their shopping cart. A customer will be able to do this for different plays.		
Preconditions:	A customer must have a registered account with the booking system.		
Postconditions:	A customer can proceed to checkout		
Priority:	High		
Frequency of use:	High		
Normal Course of Events:	A customer will be able to add several seats associated with any show to their shopping cart.		
Alternative Courses:	If the customer does not have an account, the system will redirect them to create an account to purchase tickets and explain why this is necessary.		
Exceptions:	None		
Includes:	None		
Special Requirement	A customer needs an account with the system.		
Assumptions:	A user will have already a registered account with the system.		
Notes and Issues:	N/A		

Use Case ID:	003		
Use-Case Name:	Customer Completes Purchase		
Created By:	Justyn Rippie	Last Updated By:	
Date Created:	2/23/2022	Date Last Updated:	
Actors:	Customers		
Description:	Once a customer has selected the seats for the plays they wish to attend, they complete their purchase by clicking check out in their shopping cart. The user will then provide a payment method (credit card) to complete their purchase.		
Preconditions:	A user must have a preexisting account.		
Postconditions:	User's cart will be emptied, and seats will be sold.		
Priority:	High		
Frequency of use:	High		
Normal Course of Events:	A normal check out will result in a successful transaction of the customers seats for the play(s). The system will notify the user of the successful transaction and email them a receipt of their purchase and information the seats and play. The users shopping cart will then be cleared.		
Alternative Courses:	If the payment method is denied, the system should alert the user if the transaction fails and not complete the purchase and allow them to try to enter their payment method again. All items in the cart should remain.		
Exceptions:	None		
Includes:	None		
Special Requirement	None		
Assumptions:	The user will already have an existing account		
Notes and Issues:	N/A		

Use Case ID:	004		
Use-Case Name:	Theater Admin can create play		
Created By:	Justyn Rippie	Last Updated By:	
Date Created:	2/26/2022	Date Last Updated:	
Actors:	Theater Admin		
Description:	A theater admin will log into the administrator side of the system. They will then be able to select an option to create a new play and edit the following: show times/dates, seat prices, title, description.		
Preconditions:	Must be a Theater Admin User		
Postconditions:	A new play is added to the		
Priority:	High		
Frequency of use:	High		
Normal Course of Events:	A new play will be added to the system and stored into the database. The customer will and theater admin will be able to view the newly created show.		
Alternative Courses:	If the show ID in the database already exists, the system should alert the user that the show exists and present the name and show ID.		
Exceptions:	None		
Includes:	None		
Special Requirement	None		
Assumptions:	The theater admin will be trained on how to perform this action		
Notes and Issues:	N/A		

Use Case ID:	005		
Use-Case Name:	Theater Admin Updates Show		
Created By:	Justyn Rippie	Last Updated By:	
Date Created:	2/26/2022	Date Last Updated:	
Actors:	Theater Admin		
Description:	A theater admin will log in into the Administrator side of the system and have an option to update any information for any show. This will include updating: show times/dates, seat prices, title, description		
Preconditions:	Must be a Theater Admin User		
Postconditions:	A show will update the selected information the theater admin updates		
Priority:	High		
Frequency of use:	Medium		
Normal Course of Events:	The selected show fields will be updated from what the theater admin changes. The system should ask the user if they are sure they want to update before database is written with new information.		
Alternative Courses:	If any action fails, the system should not update the database and keep the old information.		
Exceptions:	None		
Includes:	None		
Special Requirement	None		
Assumptions:	A theater admin will have training on this selected action		
Notes and Issues:	N/A		

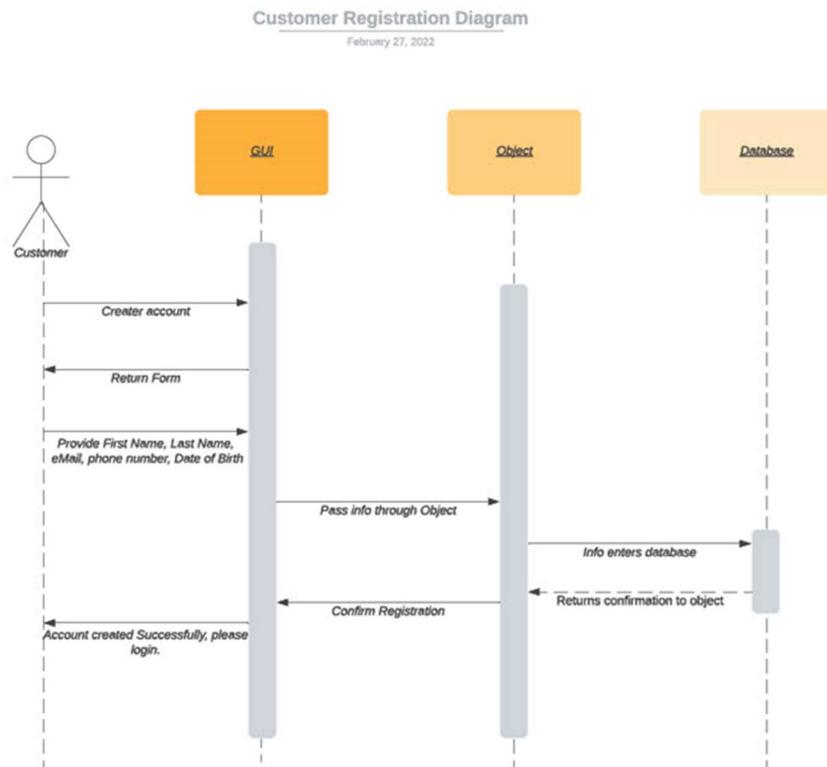
Use Case ID:	006		
Use-Case Name:	Theater Admin Deletes Show		
Created By:	Justyn Rippie	Last Updated By:	
Date Created:	2/26/2022	Date Last Updated:	
Actors:	Theater Admin		
Description:	A theater admin will have the option to delete a show. The delete option will remove the show from the system and the database.		
Preconditions:	Must be a Theater Admin User		
Postconditions:	A show will update the selected information the theater admin updates		
Priority:	High		
Frequency of use:	Medium		
Normal Course of Events:	The system will locate the show name and remove it from the system. The theater admin will receive a warning they're about to delete and ask if they want to proceed. If yes, the system will remove the show; the theater admin will receive a notification the show was removed. If the theater admin selects no, the system will not remove the show.		
Alternative Courses:	If the system cannot locate or delete the show, the system should do nothing and inform the theater admin that deletion cannot be performed, or the show name cannot be found.		
Exceptions:	None		
Includes:	None		
Special Requirement	None		
Assumptions:	A theater admin will have training on this selected action		
Notes and Issues:	N/A		

Use Case ID:	007		
Use-Case Name:	Theater Admin Generates Sales Script		
Created By:	Justyn Rippie	Last Updated By:	
Date Created:	2/26/2022	Date Last Updated:	
Actors:	Theater Admin		
Description:	A theater admin has an option to generate a report on ticket sales. The report will include the show name. The report will tell the theater admin the number of seats sold and the revenue generated. They should also see how many tickets are left.		
Preconditions:	Must be a Theater Admin User		
Postconditions:	The report will be generated and presented to the theater admin.		
Priority:	High		
Frequency of use:	High		
Normal Course of Events:	The report will list the show name, number seats sold, the total amount revenue generated for the show, and the number of seats left. The report should include the seat number sold and seat number left.		
Alternative Courses:	If the report cannot be generated, the system should alert the user with an error to report back to vendor so they may contact support.		
Exceptions:	None		
Includes:	None		
Special Requirement	None		
Assumptions:	A theater admin will have training on this selected action		
Notes and Issues:	N/A		

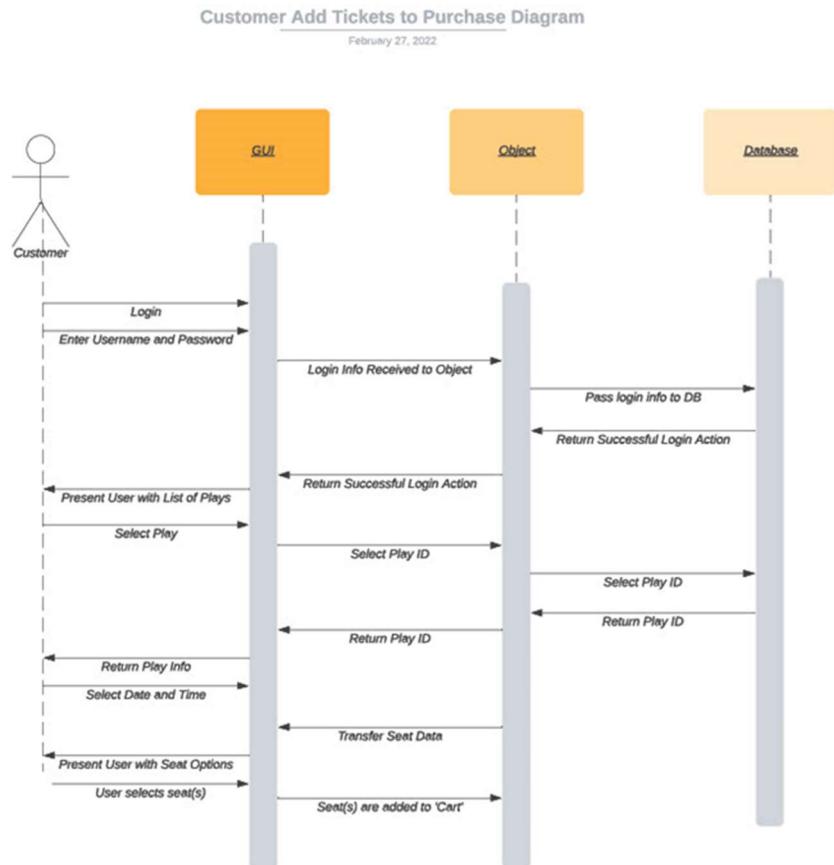
## 8.2 Sequence Diagrams

Below are listed the sequence diagrams for the interactions between the user and system and excepted outcomes.

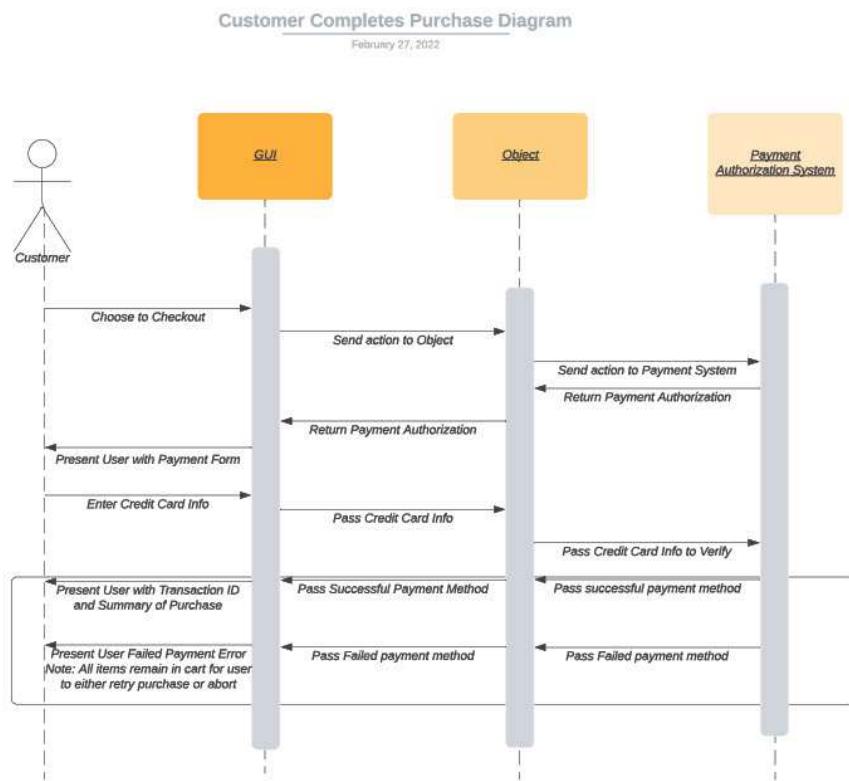
### 8.2.1 Customer Registration



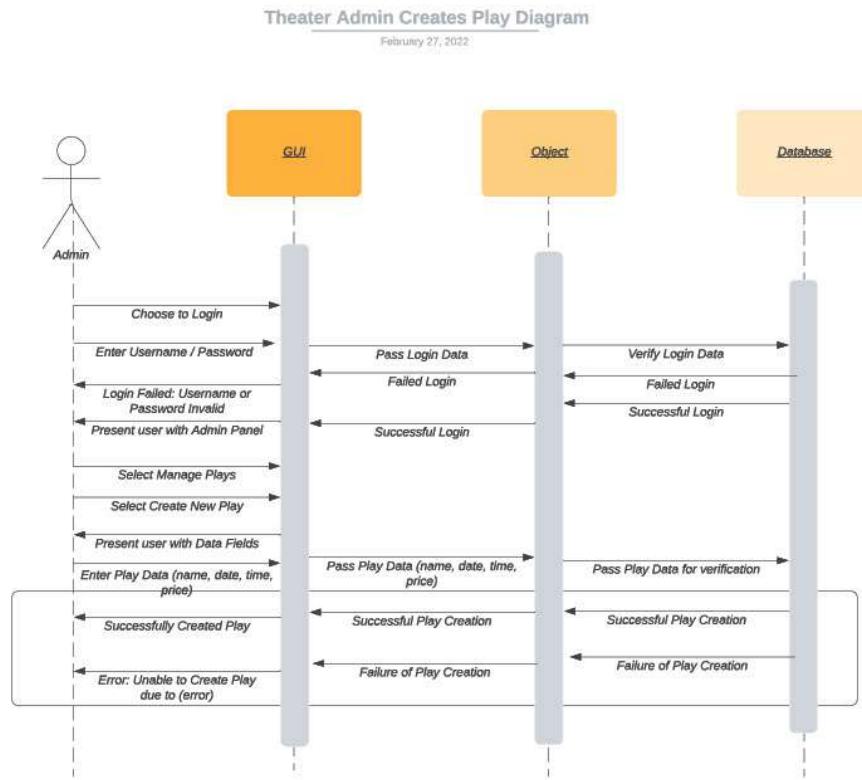
## 8.2.2 Customer Adds Ticket



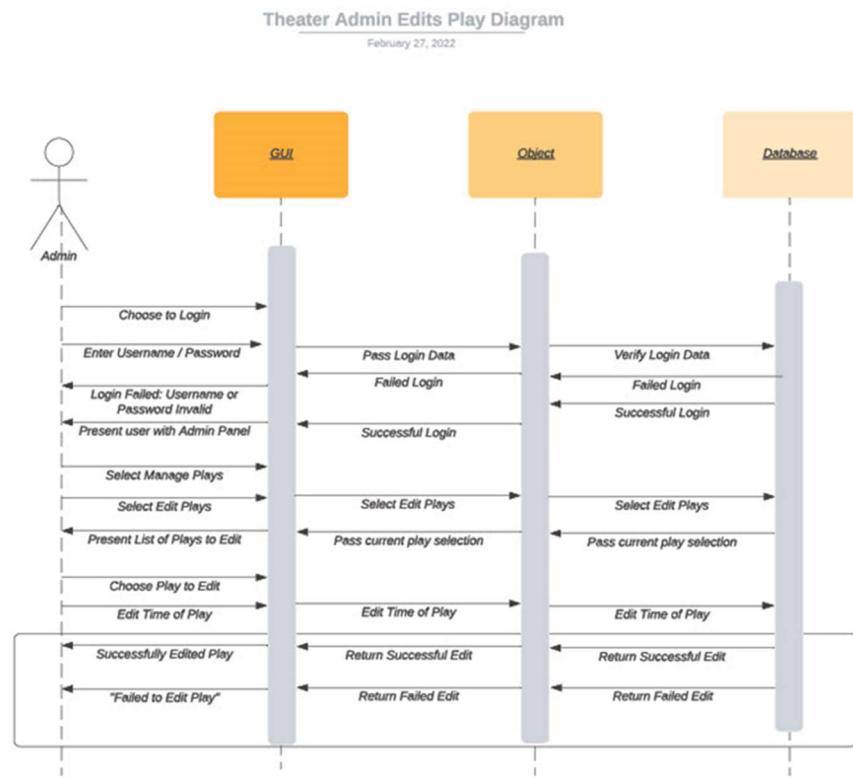
### 8.2.3 Customer Completes Purchase



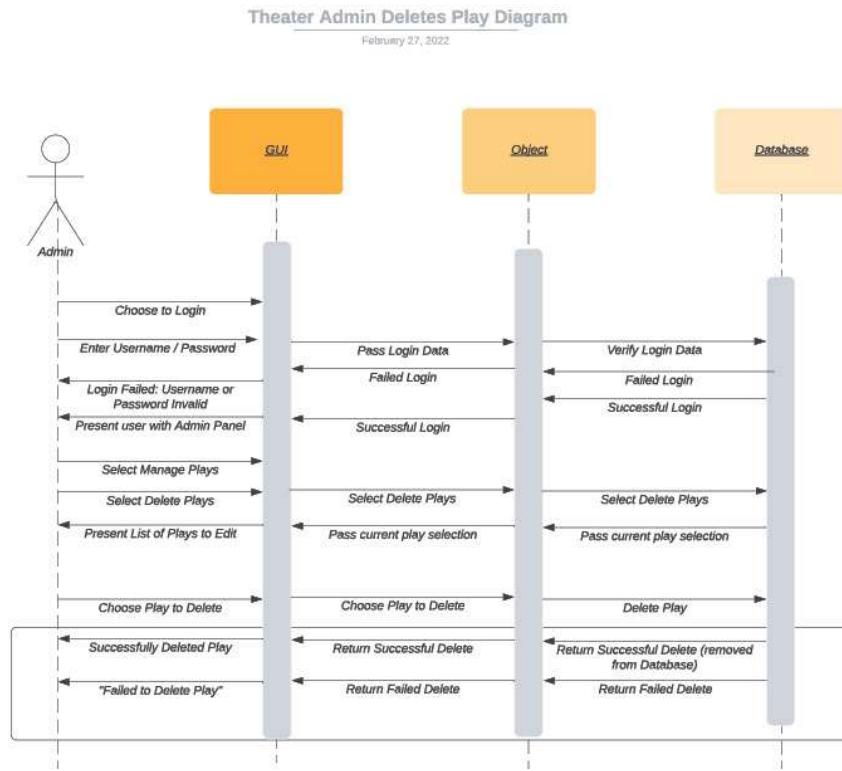
## 8.2.4 Theater Admin Creates Play



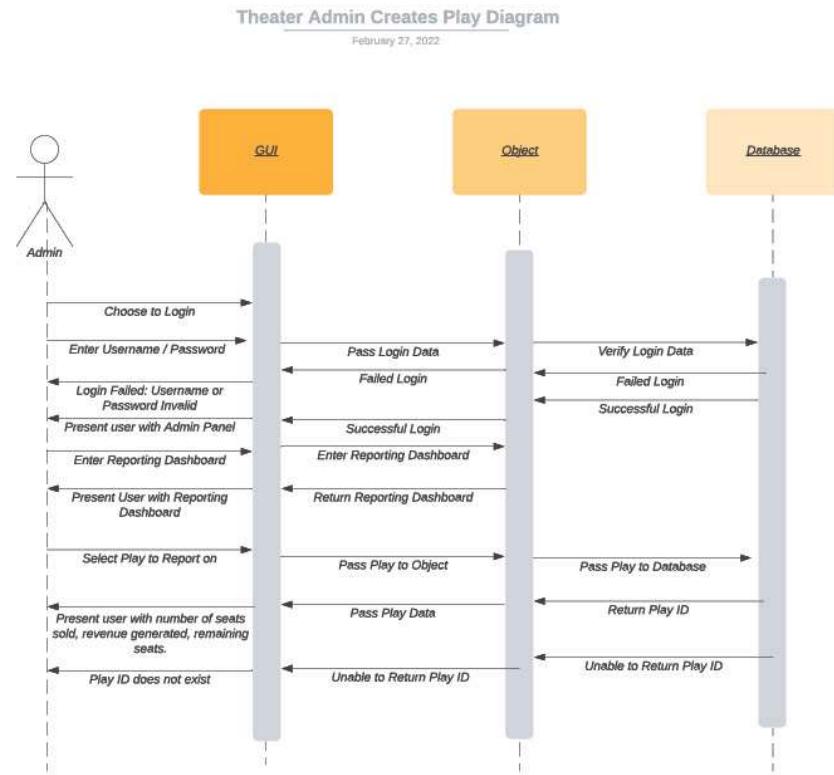
## 8.2.5 Theater Admin Edits Play



## 8.2.6 Theater Admin Deletes Play



## 8.2.7 Theater Admin Creates Play



## 9. Non-Functional Requirements

### 9.1 Performance Requirements

Interaction with the system from either the Administrator or Customer will not take significant amounts of time. This will include when the customer is purchasing tickets or modifying their account information. It will also allow the system administrator to effectively update play information seamlessly.

### 9.2 Security Requirements

The system will protect customer account information:

- Credit/Debit Card Information
- Addresses
- Email
- Full Name
- Date of Birth
- Username and Password

The software should adhere to C#/.NET Key Security Concepts provided by Microsoft at:

<https://docs.microsoft.com/enus/dotnet/standard/security/key-security-concepts>

### 9.3 Software Quality Attributes

**Availability:** The web application will be running 24/7-365

**Reliability:** Dependent on cloud hosting provider (should have no less than 5% downtime)

**Portability:** With this being a web application, it does not require on-prem software installation (i.e., no executable files).

**Usability:** Will provide a simplistic interface for both Admins and Customers

### 9.4 Business Rules

#### Customer:

- Logging into the system
- Creating a new account
- Purchasing tickets
- Browse available plays

#### System Admin:

- Creating a show
- Updating a show
- Canceling a show
- Generating a sales report
- Logging into the system
- Creating a new Admin account

## 10. Software Testing Plan

### 10.1 Testing Types

#### 10.1.1 Data and Database Integrity Testing

Test Objective:	Ensure database access methods accurately move data from Client side to appropriate Server sources.
Technique:	<ul style="list-style-type: none"><li>Attempt to connect to Database upon program execution.</li><li>Create, Update and Delete User and Admin Accounts, verify correct columns are modified.</li></ul>
Completion Criteria:	All database modification statements function without any data corruption.

#### 10.1.2 System Testing

Test Objective:	Ensure all interfaces display and route correct data for accurate representation of client and server-side data
Technique:	<ul style="list-style-type: none"><li>Navigate to the Account modification pages and attempt to submit changes to database.</li><li>Navigate to Play as an Admin to attempt to modify Play data.</li><li>Navigate to Seat Editing page as an Admin to attempt to modify Seat data (i.e., Seat Cost).</li><li>Query database to QC the modified data</li><li>Add Items to Shopping Cart and run through Check-Out process.</li><li>Ensure Sales Report query returns and display correct information.</li></ul>
Completion Criteria:	All interfaces display and accurately transfer data between Views correctly. Account modification interfaces accurately update database. Admin sales report shows correct earnings and sales information.

#### 10.1.3 User Interface Testing

Test Objective:	Ensure the website is easily navigable and webforms are easily followed and properly filled out through form validations.
Technique:	<ul style="list-style-type: none"><li>Structure forms in a vertical manner to guide and demonstrate to the users the correct flow of data entry.</li><li>Create validations on forms to ensure all required data are captured.</li><li>Create tool tips as necessary for users to interact while filling out forms</li><li>Seek to reduce the amount of clutter in Shopping Cart to effectively display what the user is paying for and for how much.</li></ul>

	<ul style="list-style-type: none"> <li>Provide easy Play and Seat data modification for Admins, by only displaying the information that needs to be updated. (i.e., reduce the amount of extra cosmetic flair).</li> <li>Admin changes will also include validations on the forms to ensure all pertinent data is captured.</li> </ul>
Completion Criteria:	All webpages will be structured to allow all required data points to be easily entered.

#### 10.1.4 Security and Access Control Testing

Test Objective:	Verify Role based access dynamically displays the correct screen for Customer and Admin roles.
Technique:	<ul style="list-style-type: none"> <li>Create dummy Customer and Admin accounts to see the different webpage displays.</li> <li>Ensure the Admin(s) are the only users with the highest authority.</li> <li>Review and Verify the login pages route the accounts to the correct screen for appropriate account modification.</li> </ul>
Completion Criteria:	Each User Role type only displays information that is pertinent to them. Admin accounts have the highest authority.

### 10.2 Suspension Criteria and Resumption Requirements

This section will specify the criteria that will be used to suspend all or a portion of the testing activities on the items associated with this test plan.

#### 10.3 Suspension Criteria

Testing will be suspended if the incidents found will not allow further testing of the system/application under-test. This will include if neither the User nor Admin are able to modify their respective accounts or are unable to successfully login. Also included in this test is to verify the website if functional for ticket purchase. If testing is halted, and changes are made to the hardware, software, or database, it is up to the Testing Manager to determine whether the test plan will be re-executed, or part of the plan will be re-executed.

#### 10.4 Resumption Requirements

Resumption of testing will be possible when the functionality that caused the suspension of testing has been retested successfully. Changes made to the software will be documented and updated in the GitHub repository.

#### 10.5 Test Data

Test data requirements are drawn up based on the functional requirements that are due for testing. The testing team will identify test cases that can be grouped into test scenarios and detail the data required to complete the testing activities.

## 11. Software Testing Results

### 11.1 TEST SUMMARY

Testing involved accessing the web page was successful and that each of the menu buttons were

routing to the correct pages. It also included Registering a New Account, Performing Login and Logout situations and modifying account information. We have ensured that Play header information was the only information viewable if an account was not logged into. Admin account access was verified to display more functionality on the page and can Generate Reports.

**Project Name:** Los Portales Theater

**System Name:** Los Portales Theater

**Version Number:** Version 1.0

**Additional Comments:** MVC (Model View Controller) Web Application

### 11.1.1 UNIT TEST FOR PLAYSCONTROLLER

**Test Owner:** *Justyn Rippie*

**Test Date:** 02 May 2022 – 03 May 2022

**Test Results:** All Unit tests were passed during this testing period for the PlaysController. Tests were written to ensure a valid case would successfully pass. The test confirmed that the controller was indeed able to create a new play, edit an existing play, and successfully delete a play.

**Additional Comments:** One thing that was noticed during the unit testing of this controller was that the database was not receiving the write changes. However, when the controller was debugged in the development environment, all database actions of create, read, update, and delete were successful.

### 11.1.2 DATA AND DATABASE INTEGRITY TESTING

This test ensured that no database type issues arose when various Insert, Delete and Update transactions actions were performed.

**Test Owner:** Al Yazzie

**Test Date:** 05/01/2022- 05/05/2022

**Test Results:** Connection to the database succeeded. Inserting, Deleting, and Updating data all worked as intended. Data from each table was retrieved properly without issue.

**Additional Comments:** The only issue that arose was a missing column that was not present in the Play\_Seat\_Cart.cs migration. There was an additional IsSold column on the Seat table that was missing but after it was added to the migration and the database was updated it, there were no other issues pushing and pulling from the database.

### 11.1.3 SYSTEM TESTING

System Testing involved verifying New Account creation, Login/Logout processes and Account modifications worked. This testing verified Administrator level access displayed more functionality to grant them the ability to Create, Modify and Delete Play information. Lastly, it verified the correct data was returned when a Report was generated.

**Test Owner:** Al Yazzie

**Test Date:** 05/01/2022- 05/05/2022

**Test Results:** All account creation procedures performed passed and data was inserted into the database correctly. New customer registration page wrote to the database and all validations worked and prevented users from skipping over required fields. Admin account was able to successfully perform CRUD operations on Play and Seat objects. Generating Reports as an Admin successfully returned correct data as intended. Queried database as a

method to compare results. Tickets from multiple Plays were successfully added to the Cart without issues and turned seat color red to represent the seat was sold. No issues were present with Checkout procedures and sold seat data was correctly written to the database.

**Additional Comments:** An issue arose where an Admin was unable to create a new Play, this was due to the IsSold column missing in the migration as described in the Data and Database Integrity Testing section. The issue was resolved and allowed the admin account to function normally.

#### 11.1.4 USER INTERFACE TESTING

This test verified all the pages were routed correctly and the correct methods to update data in the database were functional.

**Test Owner:** Al Yazzie

**Test Date:** 05/01/2022- 05/05/2022

**Test Results:** All outlined tests passed. All links and menu buttons on the Homepage have been correctly routing to correct webpages. Menu names correctly describe where the page is going to navigate to and there are no routing issues present. Validations on user creation function. Validations for Play creation also function correctly. Admin account can successfully Create, Delete, and Edit both Play and Seat information.

**Additional Comments:** Data was reviewed using two methods: SQL query in SQL Server Management Studio and SQL Server Object Explorer embedded within Visual Studio.

#### 11.1.5 SECURITY AND ACCESS CONTROL TESTING

Testing validated Account Roles only displayed data pertinent to each account type which included the ability for Admins to see more menu functions on the homepage.

**Test Owner:** Al Yazzie

**Test Date:** 05/01/2022- 05/05/2022

**Test Results:** All tests passed. This includes the Customers ability to modify their account information (Username and Password). Ensured Customer accounts do not have the ability to create or modify Play objects. The screen Customers are presented differs from what an Admin sees. An Admin has more menu options on the top menu bar. The Admin account does have the ability to create other Admin accounts from the front-end.

**Additional Comments:** Profile and Password Update previously did not easily display Current Password and Current Username, but this was quickly rectified. Initial tests uncovered an Admin's inability to create other Admin accounts due to a case-sensitive field (Role) on the Admin Creation screen. This was fixed when we changed the font size to lower case to handle any variation of typing the word 'Admin'.

### 11.2 TEST ASSESSMENT

The testing was thorough and accounted for entering different data types in number fields to ensure there were no data conversion issues. With the implementation of required fields on forms it prevented bad data from being written to the database. Lastly, the review of the data after each functionality test ensured the data was being written and read correctly.

### 11.3 TEST RESULTS

All the tests were successful at the end of testing. There were a few issues encountered due to

columns missing in the database on the Seats table. Once that was rectified, no other issues arose when Seat objects were selected. One other issue we noticed, around Account Modifications from a Customer's perspective, was the inability to correctly identify what the contents of the combo box should be. Fortunately, this was a simple front-end font color change. Lastly, we discovered Admin Accounts were unable to create other Admin accounts. This was due to a case-sensitive attribute on the Admin Creation page. We fixed this problem by modifying the inputted data to lowercase to handle any variation of typing 'Admin' in the Role box.

## 11.4 SUGGESTED ACTIONS

Based on the results from various scenarios accounted for during the testing, no changes are required for this product. If further testing is to be conducted, the focus should be around ensuring the Cart empties upon Successful Checkout.

## 11.5 Test Summary Report Approval

The undersigned acknowledge they have reviewed the **Los Portales Test Summary Report** and agree with the approach it presents. Changes to this **Test Summary Report** will be coordinated with and approved by the undersigned or their designated representatives.

Signature: \_\_\_\_\_ Date: 05/11/2022  
Print Name: Justyn Rippie  
Title: Software Developer  
Role: Tester

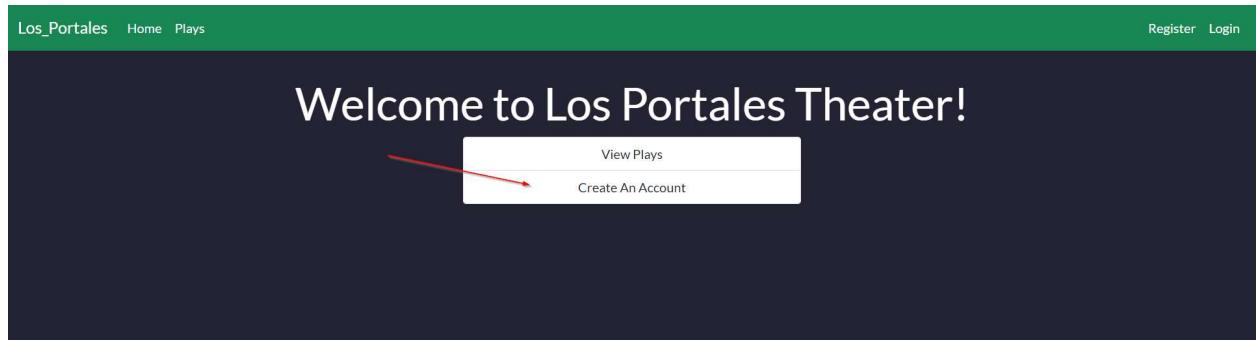
Signature: \_\_\_\_\_ Date: 05/11/2022  
Print Name: Al Yazzie  
Title: Software Developer  
Role: Tester

## 12. User Documentation

### 12.1 User / Customer Guide

#### 1. Create Account

- a. In order to create an account as a user, navigate to  
<https://localhost:7012/>
- b. Click “Create an Account” as shown below:

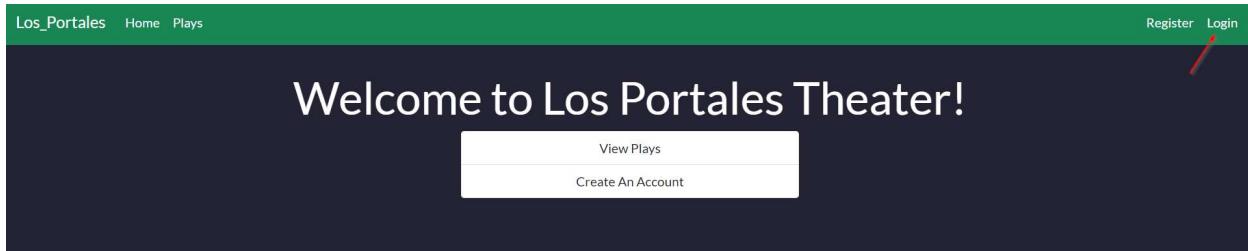


- c. Enter the relevant information on the following page, and click “Register”

A screenshot of a registration form titled 'Register' with the subtitle 'Create a new account.' The form consists of several input fields: FirstName, LastName, DateOfBirth (with a placeholder 'mm/dd/yyyy' and a calendar icon), User Name, PhoneNumber, Email, Password, and Confirm password. At the bottom is a large green 'Register' button.

**2. To login to an already existing account:**

- d. Navigate to <https://localhost:7012/>
- e. Click “Login”

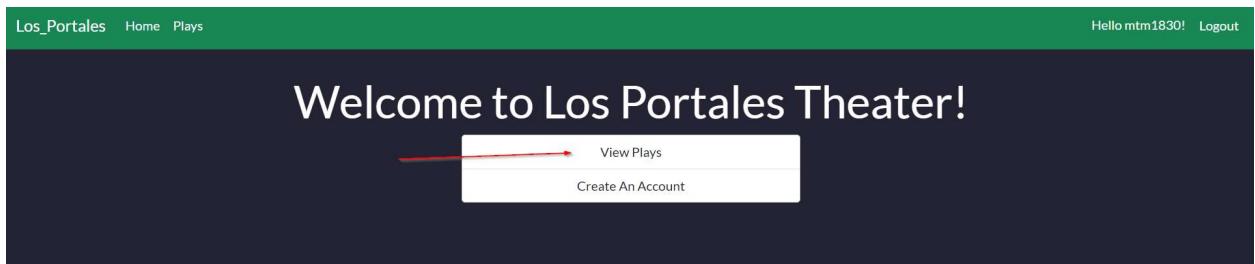


- f. Enter your username and password on the following page, and click “Login”

A screenshot of the Los Portales Theater login page. The title "Log in" is at the top. Below it is the instruction "Use a local account to log in.". There are two input fields: "User Name" and "Password", both of which have red error messages: "The User Name field is required." and "The Password field is required.". Below the fields is a checkbox labeled "Remember me?". A large green "Log in" button is centered below the fields. At the bottom, there are links for "Forgot your password?" and "Register as a new user".

- g. If you have forgotten your password, click on “Forgot your password?”
- h. If you do not have an account, click “Register as a new user”

- i. You can click “remember me” if you would like the site to remember your information for future login attempts.
2. To view the available plays, whether logged in our not, click “View Plays”

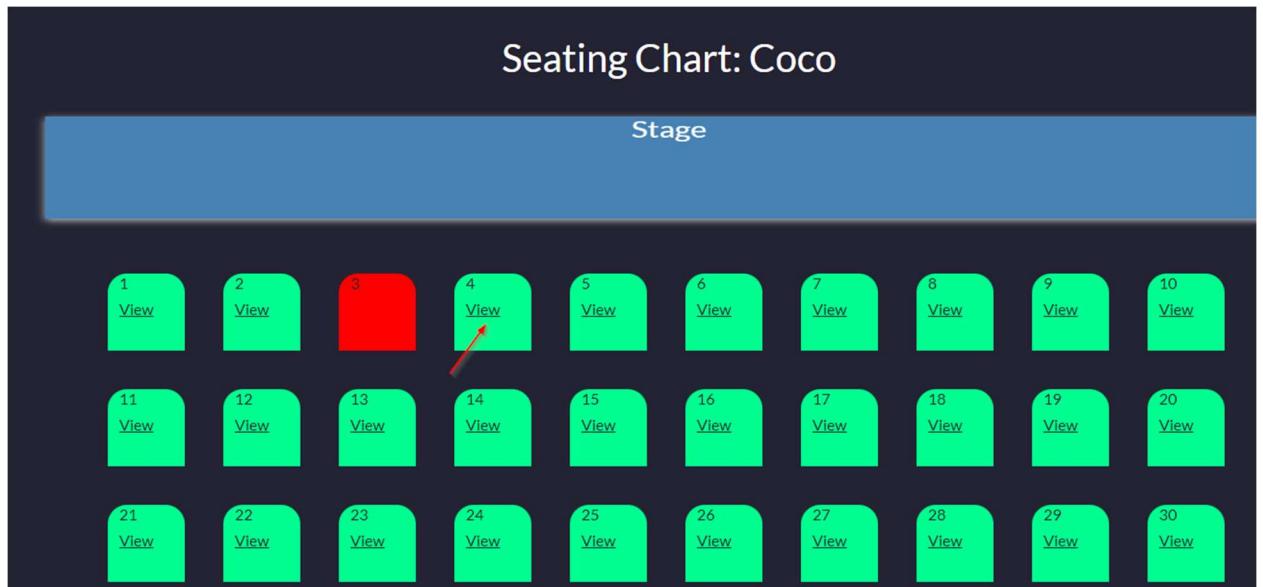


3. To purchase tickets for a play:

- a. Login to your account.
- b. Select a play you want to buy tickets for
- c. Click “Seats”

Current Plays			
PlayName	PlayDate	PlayTime	
Coco	8/8/2022	1:00 PM	<a href="#">Seats</a>
Soul	6/10/2022	1:00 PM	<a href="#">Seats</a>
Soul	6/10/2022	7:00 PM	<a href="#">Seats</a>

- d. Choose the seat(s) you want by clicking “View” and then “Add to Cart”



## Add To Shopping Cart

Seat 4

SeatNumber	4
Price	\$25.00
Play	Coco

[Add to Cart](#) →

[Back to Seating Chart](#)

e. From there, you can choose to checkout

Shopping Cart			
Seat Number	Play Name	Price	Remove Item
4	Coco	25	<a href="#">Remove Item</a>
Total			
	<a href="#">Check Out</a>		<a href="#">Back to Plays</a>

f. Once you are on the checkout page, you are going to want to enter all information for your payment that is requested, and then click "Checkout"

CreditCardNumber  
SecurityCode  
Month  
January  
Year  
2022  
NameOnCard

**Check Out**

**g. After your checkout is successful, you will be presented with a receipt**

### Check Out Success

Summary of Transaction			
Transaction ID	Seats Purchased	Total Cost	
1001	1	25	

**Details Of Your Purchased Seat(s)**

Name of Play	Seat Number Purchased	Date	Time
Coco	4	8/8/2022	1:00 PM

## 12.2 Administration Instructions:

### 1. To Create a Play:

- Login as an admin user
- Navigate to “Plays” or “View Plays” as shown below:

Los Portales Home Plays Admins Generate a Report

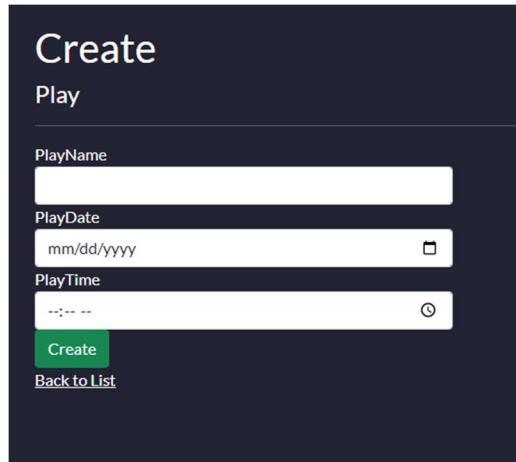
Welcome to Los Portales Theater!

**View Plays**

Create An Account

- Click on “Create New”

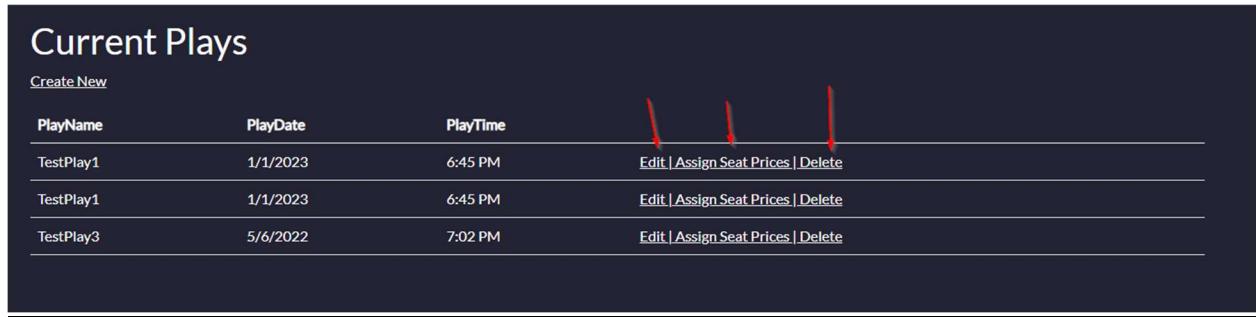
- d. Enter the relevant information for the new play, and click “create”



The image shows a dark-themed web form titled "Create Play". It has fields for "PlayName" (a text input), "PlayDate" (a date input with a calendar icon), and "PlayTime" (a time input with a clock icon). Below these is a green "Create" button and a link "Back to List".

2. To edit an existing play:

- Click on “Plays”
- Click on edit to edit the name date and time of an existing play, click “assign seat prices” to assign pricing for the play, and click “Delete” to remove the play.

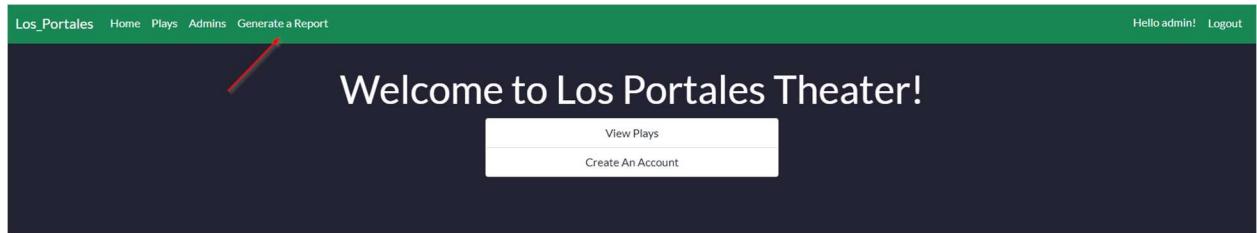


The image shows a table titled "Current Plays" with a "Create New" button. The table has columns for "PlayName", "PlayDate", "PlayTime", and actions. Red arrows point from the text "Edit | Assign Seat Prices | Delete" to the corresponding links in the table rows.

PlayName	PlayDate	PlayTime	
TestPlay1	1/1/2023	6:45 PM	<a href="#">Edit</a>   <a href="#">Assign Seat Prices</a>   <a href="#">Delete</a>
TestPlay1	1/1/2023	6:45 PM	<a href="#">Edit</a>   <a href="#">Assign Seat Prices</a>   <a href="#">Delete</a>
TestPlay3	5/6/2022	7:02 PM	<a href="#">Edit</a>   <a href="#">Assign Seat Prices</a>   <a href="#">Delete</a>

3. To run reporting on play Data:

- Click on “Generate a report”



**b. Click on the corresponding play in the dropdown menu, and then choose to report on Seats sold, or seats available, or both:**

**Generate a Report**

Plays

TestPlay3

Please select one or more the following report options

SeatsSold  
 SeatsAvailable

[Index](#)



The image shows a laptop and a tablet side-by-side on a desk. The laptop screen displays a report titled 'Report Results for TestPlay3' with a bar chart showing data for 'Sales' and a pie chart showing distribution. The tablet screen shows a calendar application. Both devices are connected to a white keyboard.

## 13. Future of the Software

The future of this software is unknown. A meeting will need to be made with the client in order to discuss further development of the software. We will maintain support of the current software. Additional features that we want to add are:

- Emailing a ticket purchase to a customer
- Verifying a user's email address by sending them an email
- Adding functionality so a user can reset their password
- Making the app mobile friendly

Likewise, a meeting with the client will need to be set in order to discuss further development of the software.