SOFTWARE DESIGN DOCUMENT

For the Los Portales Theatre Booking System

By Team: Walker Monaghan, Zachary Dowding, Vern Watkins

Table of Contents

Contents

1.0 Introduction

1.1 Goals and Objectives

The goal of our software is to enable an online system to sell seats for different plays at the "Los Portales" theatre. The customers can access this system anywhere and anytime.

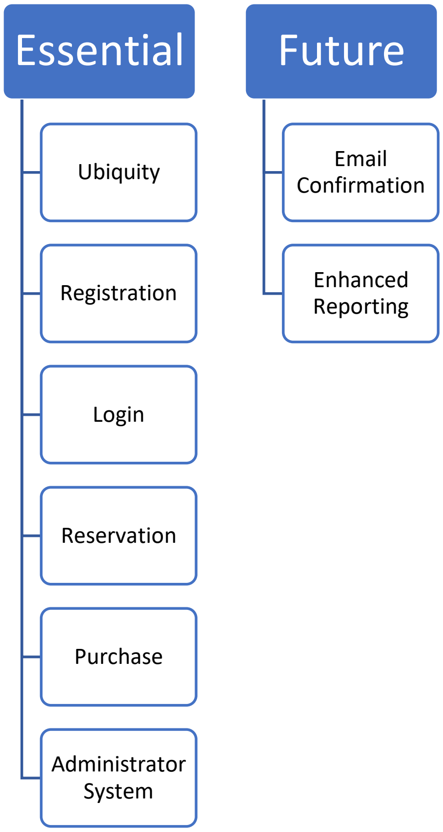
This system is designed to meet customer and administrator needs in effectively and efficiently purchasing specific seats and theater showings.

This system should empower the system administrator to quickly modify showing information (add/remove showings, change seat prices, change available seats).

The major design considerations for this proposed system are ubiquity, ease-of-use, and transaction accuracy.

1.2 Scope

Ranking of Major Processing Functionality



<u>Major In-Scope Inputs and Outputs</u>

Customer Inputs	**Customer Outputs**
Registration Information	Showing if customer registration was successful or unsuccessful.
Login	Showing if customer login was not accepted (then the software would show a message to 'try again') or accepted (then the software would enable links to the theatre pages)
Selecting a show to view available seats for that show.	Showing available seats for all shows.
Selecting and unselecting seats for a show from a page showing which seats are available for that play.	Showing which seats in all shows the customer has requested.
Entering payment information into a shopping cart page that shows all selected seats for all shows and their cost.	Showing if the customer payment information entered was not accepted (then the software would show a message to 'try again') or accepted (then the software would show a message 'Thank you for your payment').
on the shopping cart page, they can unselect seats that they have selected before (where the cost will be adjusted).	Showing which seats in all shows the customer has purchased and their total cost.

Administrator Inputs	**Administrator Outputs**
Entering administrator login information.	Showing if administrator login was not accepted (then the software would show a message to 'try again') or accepted (then the software would open the administrator page)
Adding plays with their dates and times.	Showing all shows offered.
Establishing seat pricing for all shows.	Showing available seats for all shows (the same way that the customer saw those seats).
Adjusting seat pricing for all shows.	
Making a system request to generate a report showing how many seats have been sold for a specific play and date.	Creating the administrator report that shows how many seats have been sold for a specific play and date.

The processing functionality for these inputs and outputs are essential.

Processing functionalities that are desirable are having the system send an email to the customer with their purchase details and optimizing the website for mobile devices or tablets.

2.0 Data Design

2.1 Database Description

The database used for our project is provided by www.ooowebhost.com and is a MySQL database which will have the following tables:

2.1.1 customer table

- created before website becomes ready for customer use, contains name, age, address, telephone, email. When completed (all items answered), the customer is prompted to create a login username and password which will be entered into the login table.
< For our Limited Version of the program, we will make and use just one customer.

2.1.2 login table

- contains the login username and password from a customer registering (done once for each customer)
- when the login button is clicked, a comparison of the username and password entered (in the textboxes on the welcome page) is made with its entry in the login table, and used to either verify the login information as correct and allow the customer to view seats for plays, or if the login information is not correct to ask for the correct username and password.
- created before website becomes ready for customer use and grows on a row-by-row basis.

2.1.3 plays table

contains:
- play number
- play title**
- play date**
- play times**
- play description**

(**shown on the welcome page, and in the administrator report)
- created before website becomes ready for customer (and administrator) use and grows on a row-by-row basis when the administrator adds plays.

2.1.4. template table

- has 96 rows
- to use for making copies of itself, used for making these two tables: the statusofseatsplay{play#} table as follows, and the seatspercustomer{cust#}play{play#} table as follows.

2.1.5. statusofseatsplay{play#} table

- contains the availability status of seats for a specific play, to show which seats for a play are available or have been taken when the customer goes to a play's theatre page. This table has 96 rows.
- created by the administrator from a read-only template (accessed on the administrator page) for each new play (done after
adding that play to the plays table), who then assigns *prices* for that play's seats.

- the *template* is created before website becomes ready for customer use.
< For our Limited Version of the program, we will make and use just 1 table named statusofseatsplay1 from the template.

2.1.6 seatspercustomer{cust#}play{play#} table

- created when a customer, while on a play's theatre page, requests any seats for that play. This table has 96 rows.
- after a seat request, the theatre page for that play provides the play# and the cust# of the customer accessing the theatre page to make, for example, a seatspercustomer2play2 table.
- this table contains which seats that the customer has requested for that particular play, and which seats the customer has purchased after a successful purchase.
- also contains the *prices* of seats for that play (as assigned by the administrator in the statusofseatsplay{play#} table)

When a customer requests any seats for a different play, a new seatspercustomer{cust#}play{play#} table is created (with the (play#} being different.

Used in conjunction with the above plays and customer{cust#}forplay{play#} tables to provide the required data for a customer's shopping cart of all seats requested from all plays.

Before paying, the customer can modify any seat's requested status in this table if they decide that they do not want a seat that they requested, where the customer can do this in two ways: from the theatre page for a play whose status of seats is provided by the statusofseatsplay{play#} table, or directly from their shopping cart.

The customer can delete this entire table if, while in their shopping cart before paying, the customer decides not to go to that play.

After the customer makes a purchase, this table is Updated to make the 'finalized' seats correct, and is used to Update the above statusofseatsplay{play#} table and the transaction table.

If the customer later wants additional seats in this same play, a new entry will be made in their shopping cart, and after the purchase, this table is Updated (as before) again to make the 'finalized' seats correct, and is used to Update (again) the above statusofseatsplay{play#} table and the transaction table.

2.1.7 transaction table

created before website becomes ready for customer use, to contain the name of the play(s), the date(s) and time(s), and the number of seat(s) *bought* for those plays. This table is queried by the administrator for reporting how many seats have been sold (and the money received) for a specific play and date.

Its data is provided by all the finalized[1] seatspercustomer{cust#}play{play#} tables after a purchase, with

[1] *at the time of purchase, two or more customers may have requested the same seat(s) but only the customer who pays for*

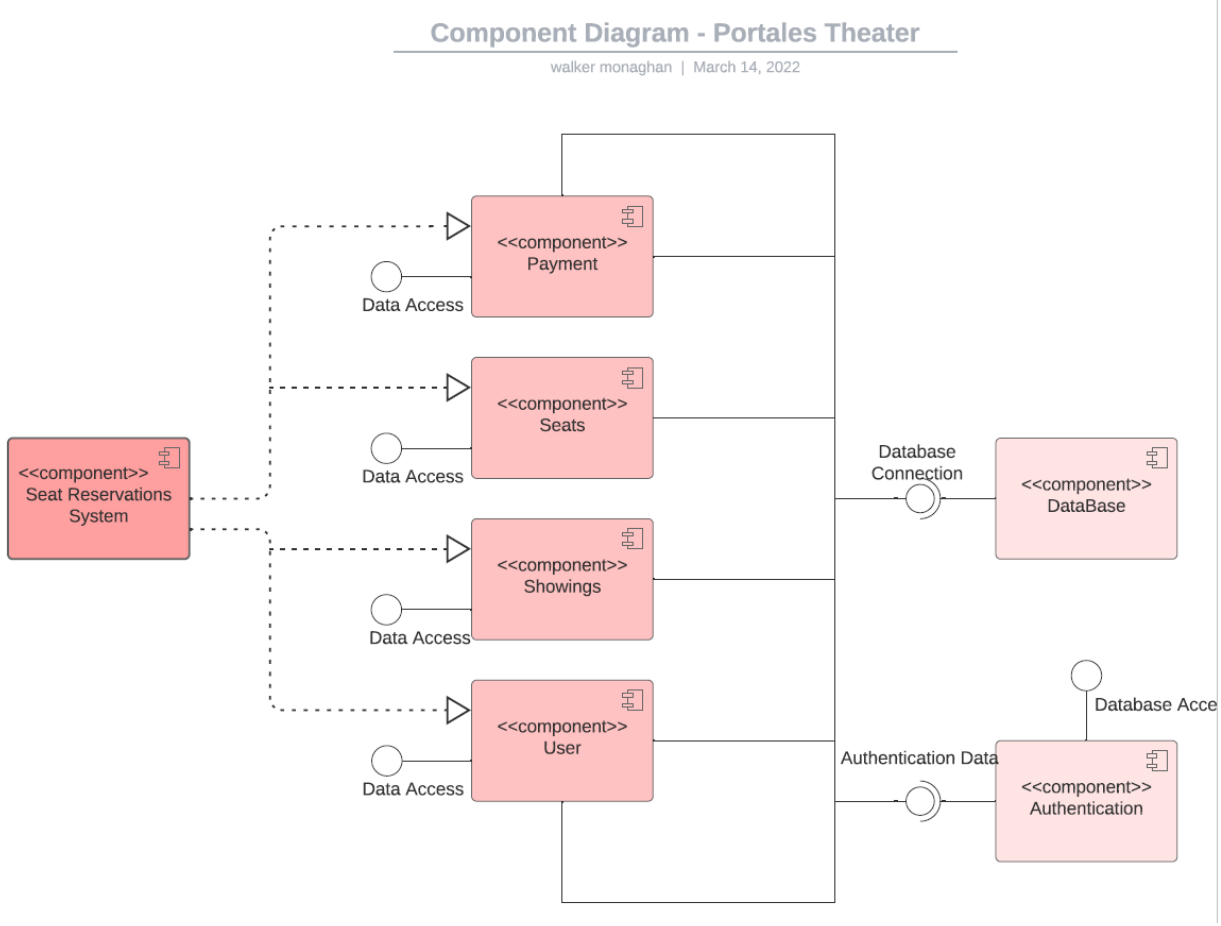
each of those tables total seats 'sum' results (for that customer) added to all the other customers sum results to produce the total number of seats that have been sold for a specific play (and date of the play from the plays table).

them first will obtain them, in that case, the other customer's purchase price paid will be adjusted and reported to them.

3.0 Architectural and Component-level Design

3.1 System Structure

3.1.1 Architecture Diagram



3.2 Decomposition Description

Component Descriptions

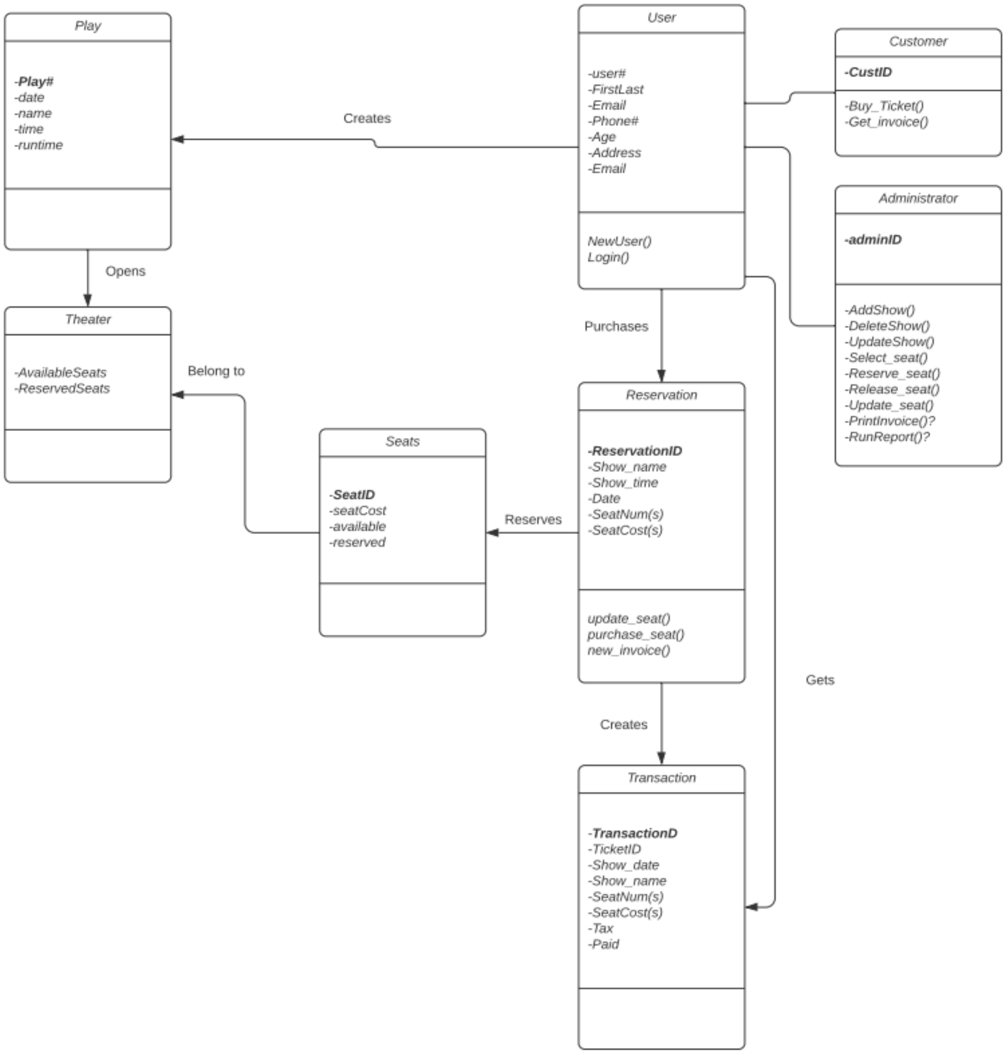
Component	Description
Seat Reservation System	This broader component houses the below subcomponents. The goal of this system is to allow users to select, reserve, and purchase seats to theater showings. This same system will allow administrators the ability to modify showings, seat pricing/availability, and to run reports on seating status.
Showings	This component controls open and available showings, which can be treated as unique events ties to a day and time. Each showing has a set of available seats at prices dictated by an administrator.
Seats	The Seats component accounts for unique seat availability, selection and de-selection, seat price, and purchased/reserved seats.

User	The user component addresses user type, customer, or administrator, as well as granting access to certain controls and information within the system as appropriate to their type.
Payment	The payment system receives, verifies, and records seat purchases.
Database	Collection of system databases with which many other system components will have access to based on method and authentication level.
Authentication	Authentication component to verify accurate User credentials, granting access to appropriate components within the overall Seat Reservation System.

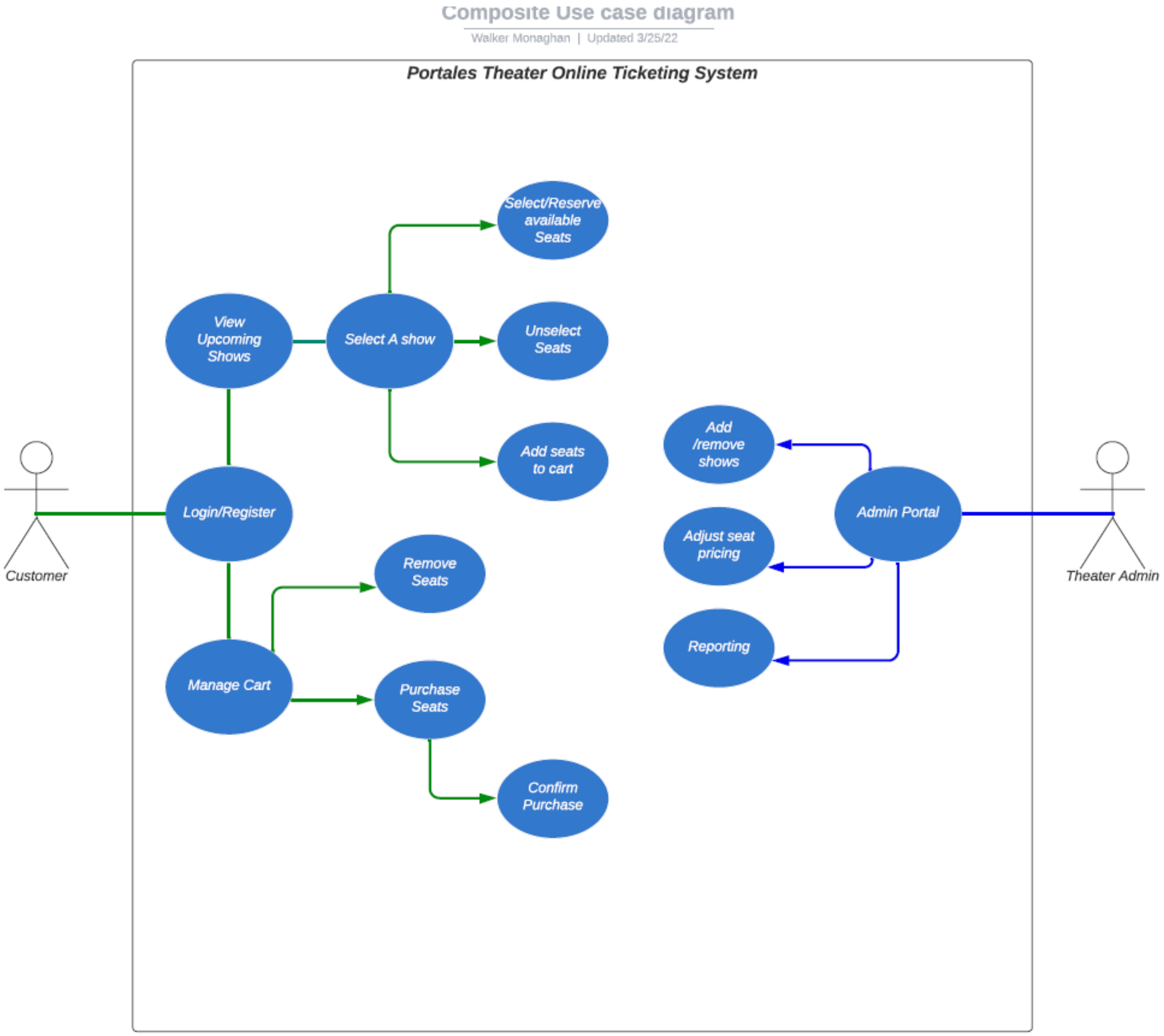
3.3 Dynamic Behavior

3.3.1 Reference Diagrams

UML Class Diagram



UML Use-Case Diagram



3.3.2 Interaction Diagrams

Sequence Diagrams for Component Use-Case scenarios.

Payment Component

Use Case - Customer – Manage Cart

Figure 1 - Remove Seats

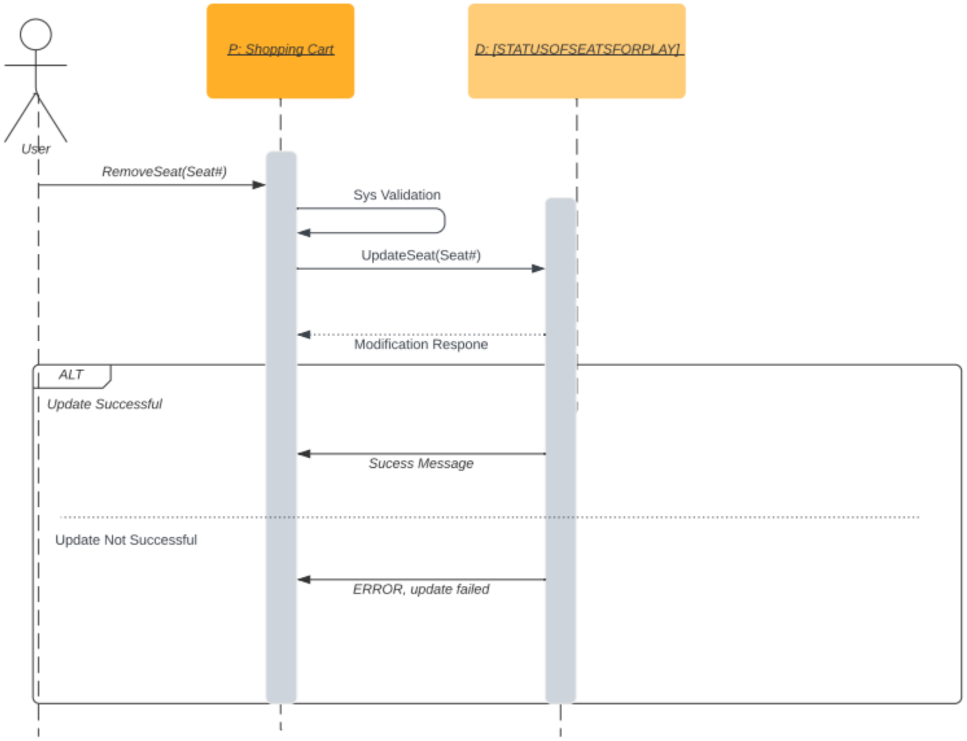
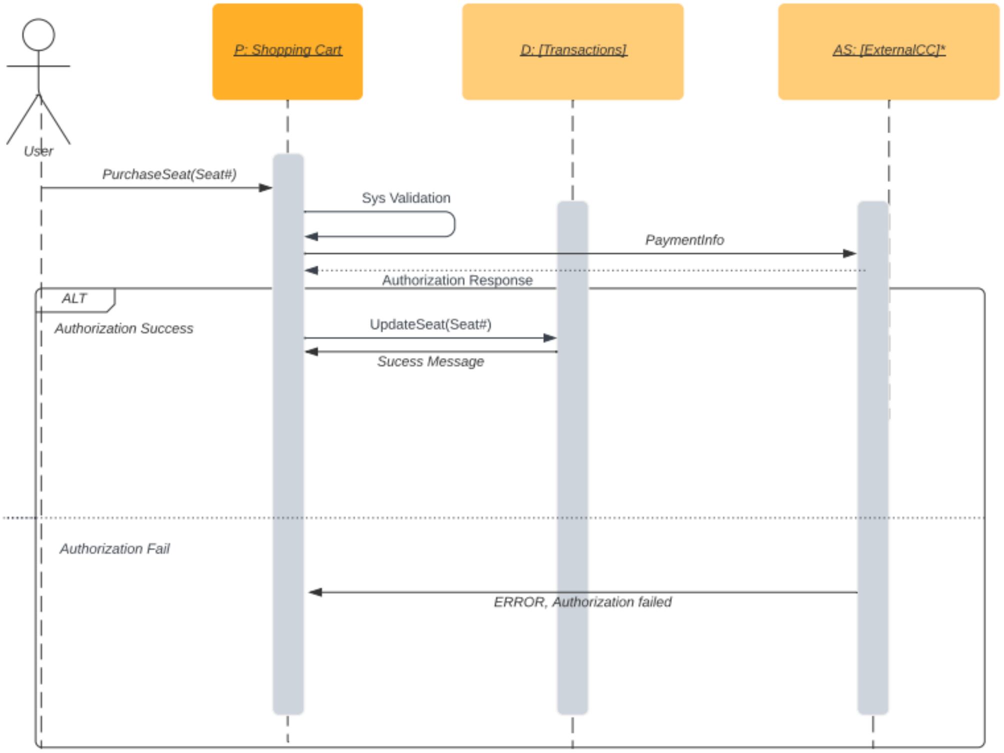


Figure 2- Purchase Seats



Seats Component

Use Case - Customer – Seat Select/Unselect

Figure 3 - Select/Unselect Seats

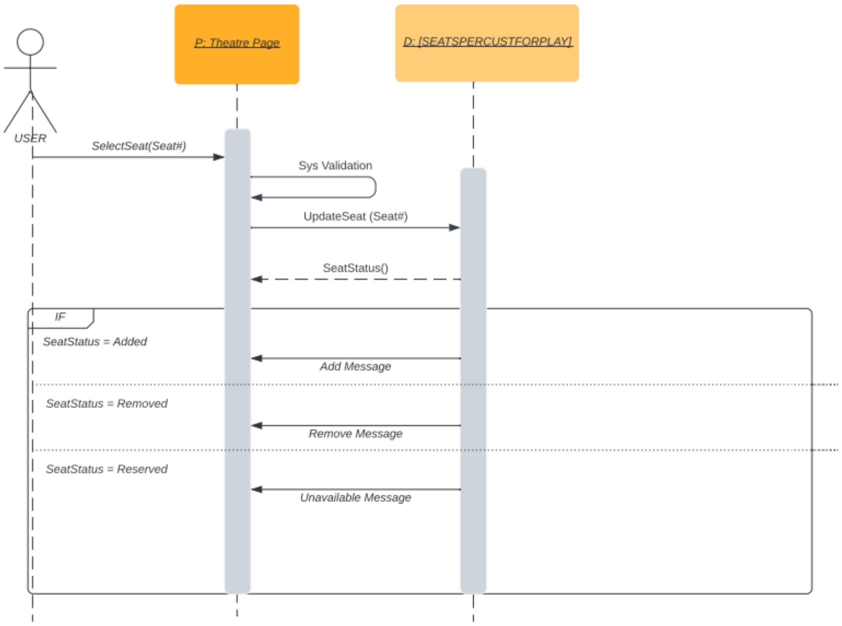
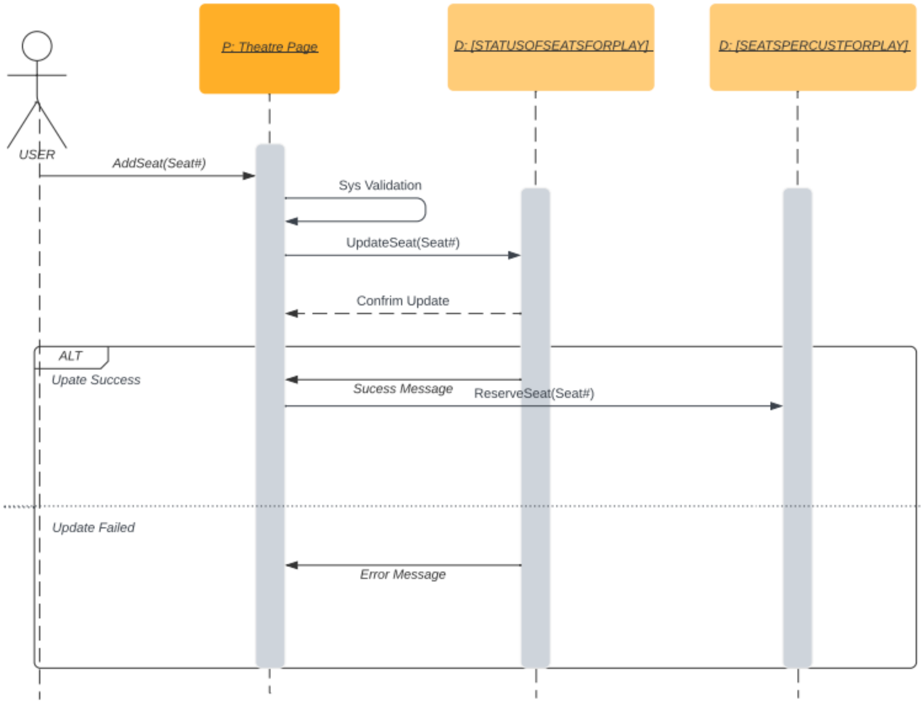


Figure 4 - Add Seat(s) to Cart



Use Case – Administrator – Adjust Seat pricing

Figure 5 - Select/Unselect (ADMIN)

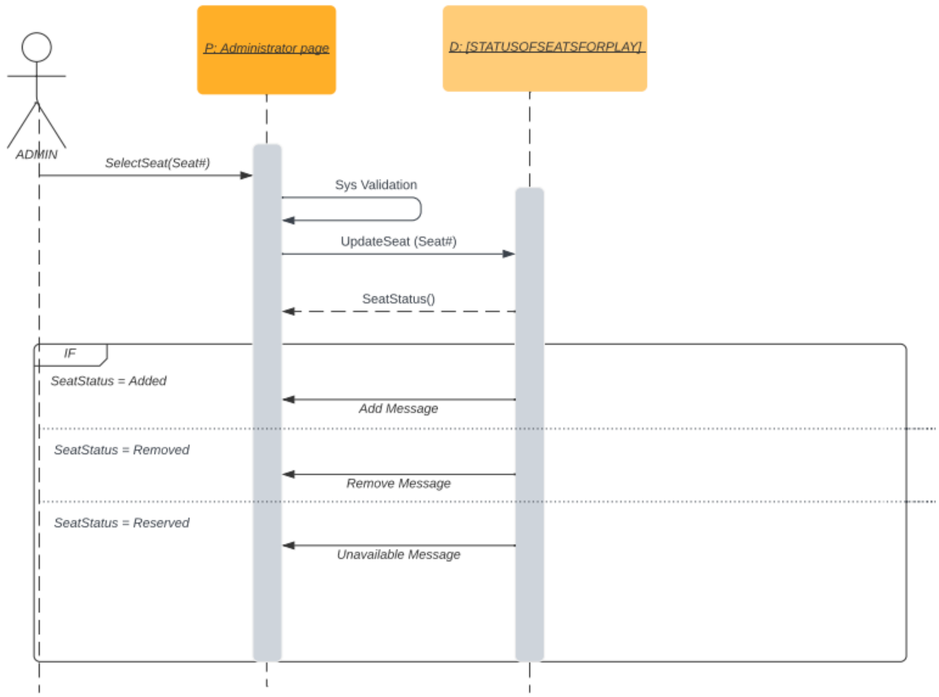
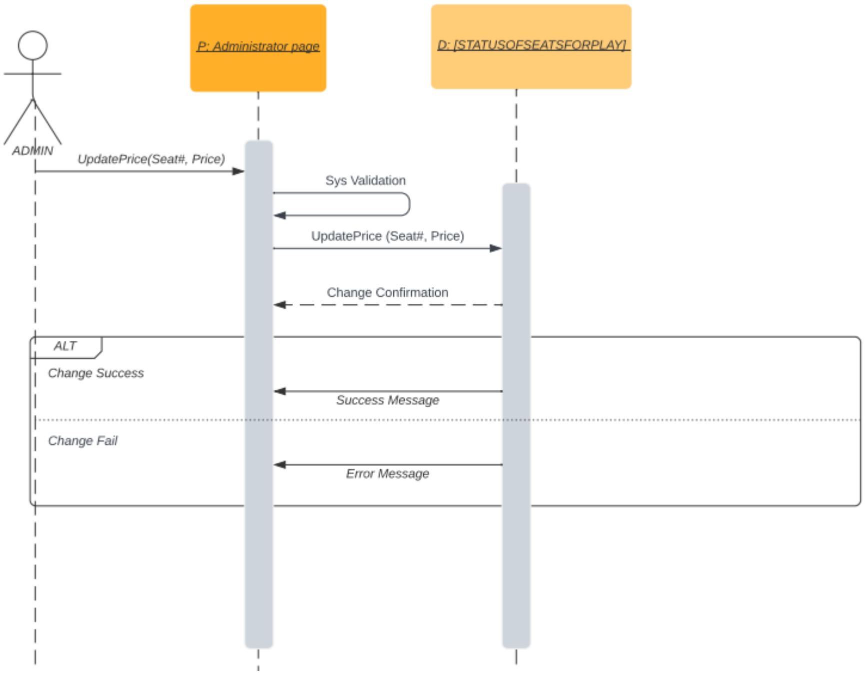


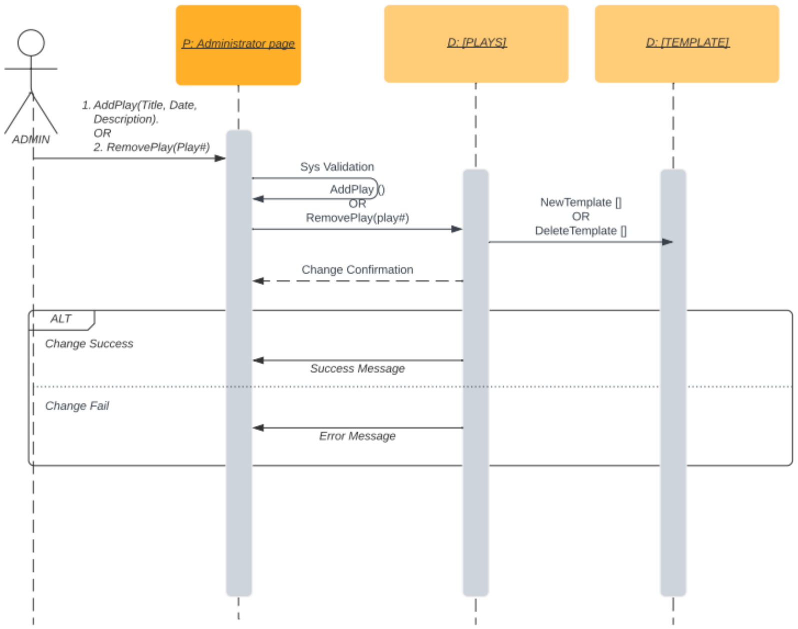
Figure 6 - Update Price



Showings

Use Case- Administrator – Add/Remove Play

Figure 7 - Add/Remove Play (ADMIN)



User

Use Case – Customer – Register

Figure 8 - Registration

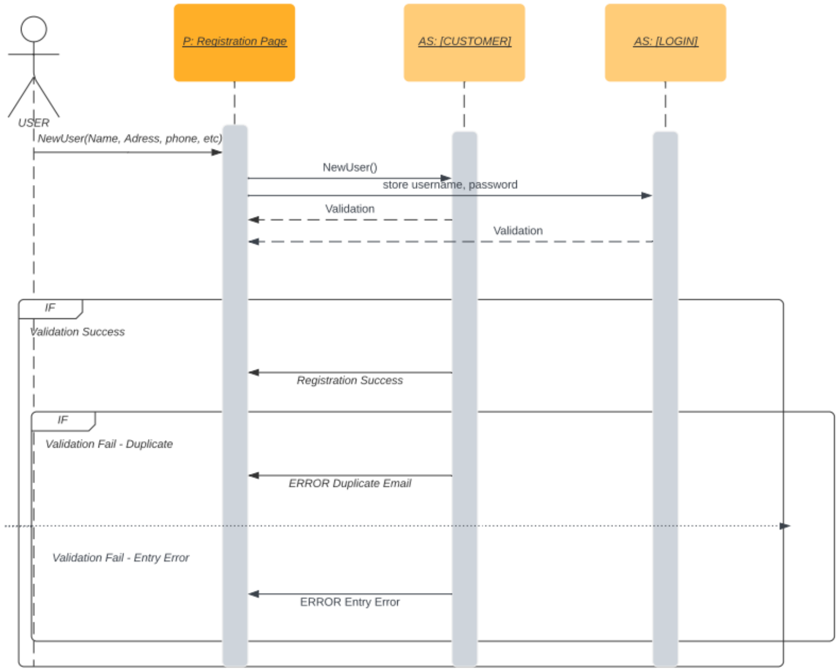
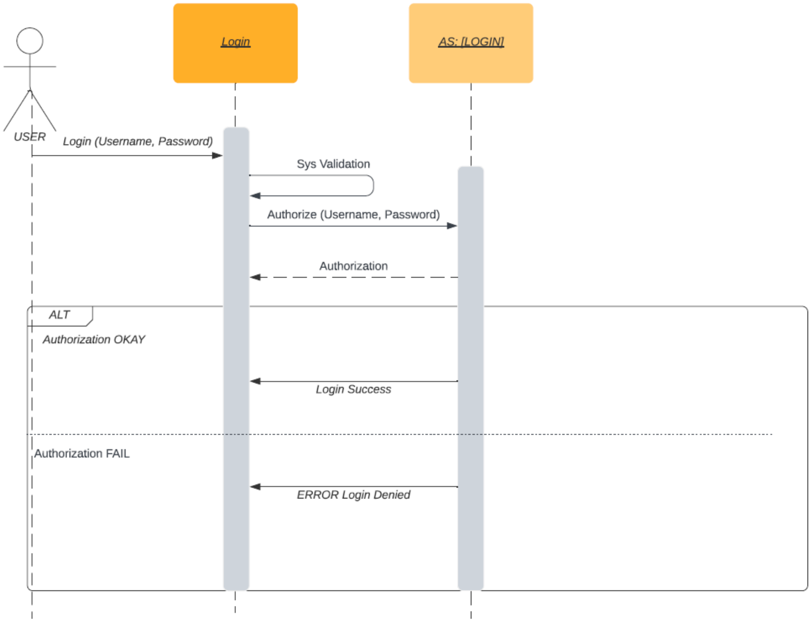


Figure 9 - Login



Use Case – Administrator

Figure 10 - Login

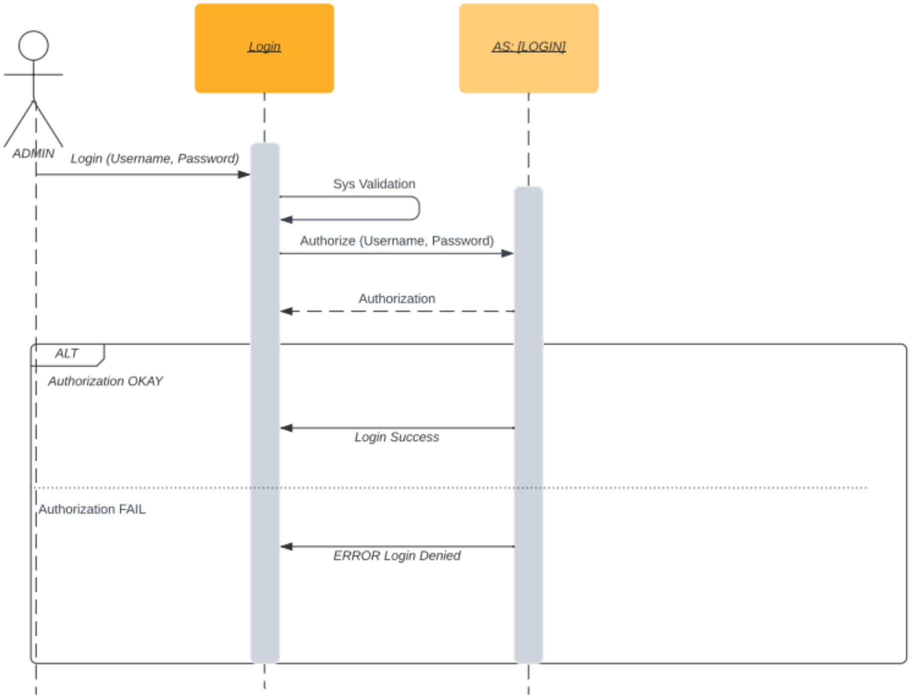
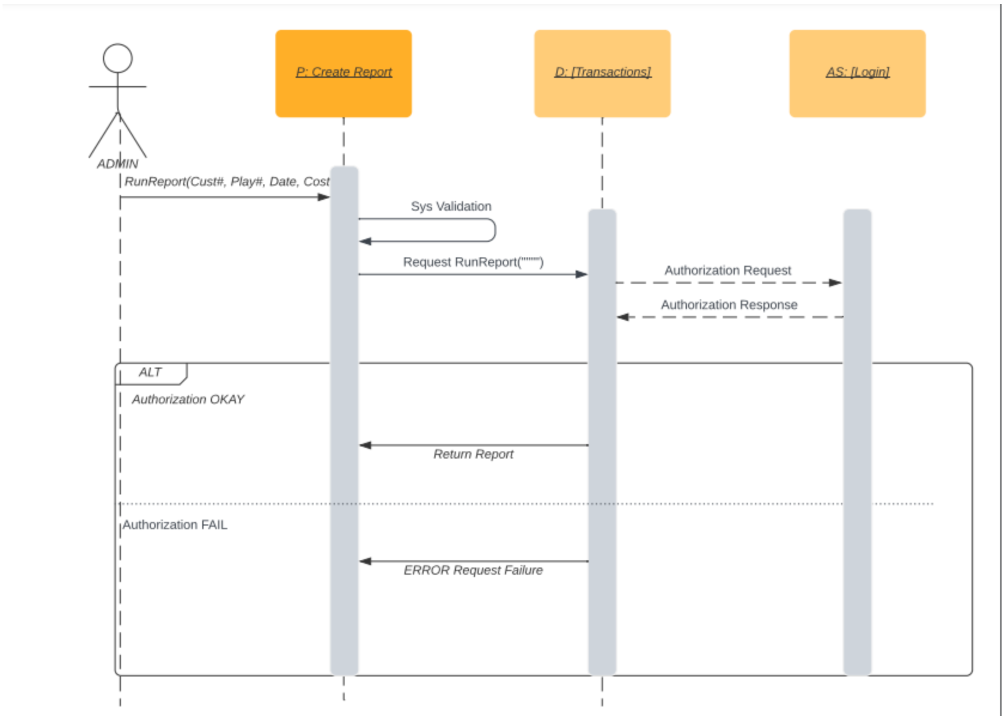


Figure 11 - Create Report

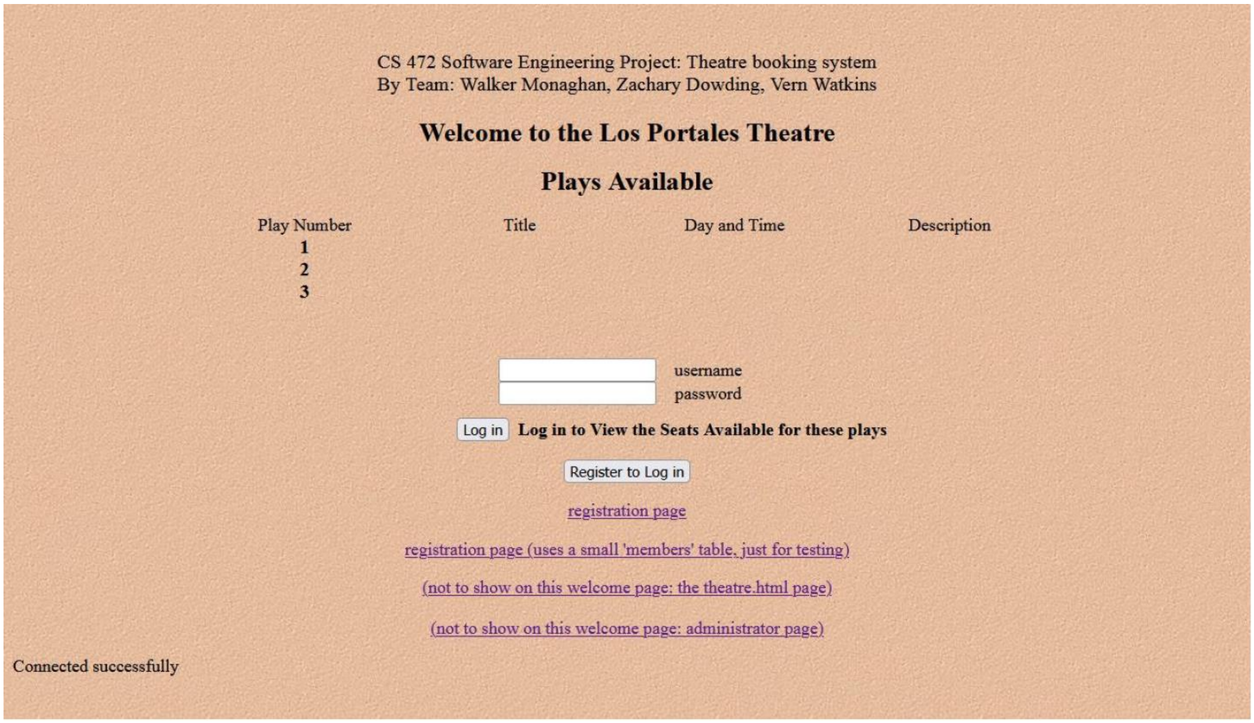


4.0 User Interface Design

4.1 Description of the user interface

4.1.1 Screen Images and descriptions

Figure 12 - Welcome Page



CS 472 Software Engineering Project: Theatre booking system
By Team: Walker Monaghan, Zachary Dowding, Vern Watkins

Welcome to the Los Portales Theatre

Plays Available

Play Number	Title	Day and Time	Description
1			
2			
3			

username
password

Log in to View the Seats Available for these plays

registration page

registration page (uses a small 'members' table, just for testing)

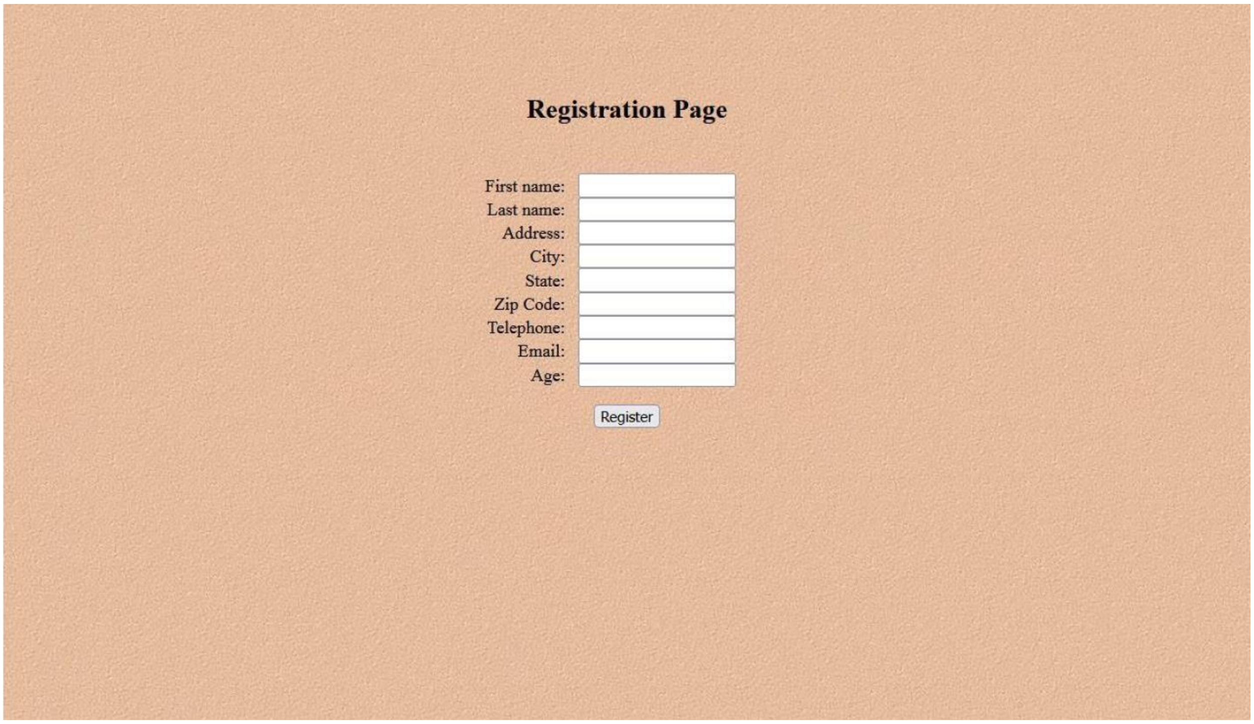
(not to show on this welcome page: the theatre.html page)

(not to show on this welcome page: administrator page)

Connected successfully

- **Shows Information about all shows offered.**
- **After registering, the customer enters their login information to be able to select a show to view available seats for that show.**
- **Shows if customer login was not accepted (then the software would show a message to 'try again') or was accepted.**
- **The Administrator enters their information and 1. Shows if it was not accepted (then the software would show a message to 'try again') or 2. Opens the administrator page.**

Figure 13 - Registration Page



Registration Page

First name:

Last name:

Address:

City:

State:

Zip Code:

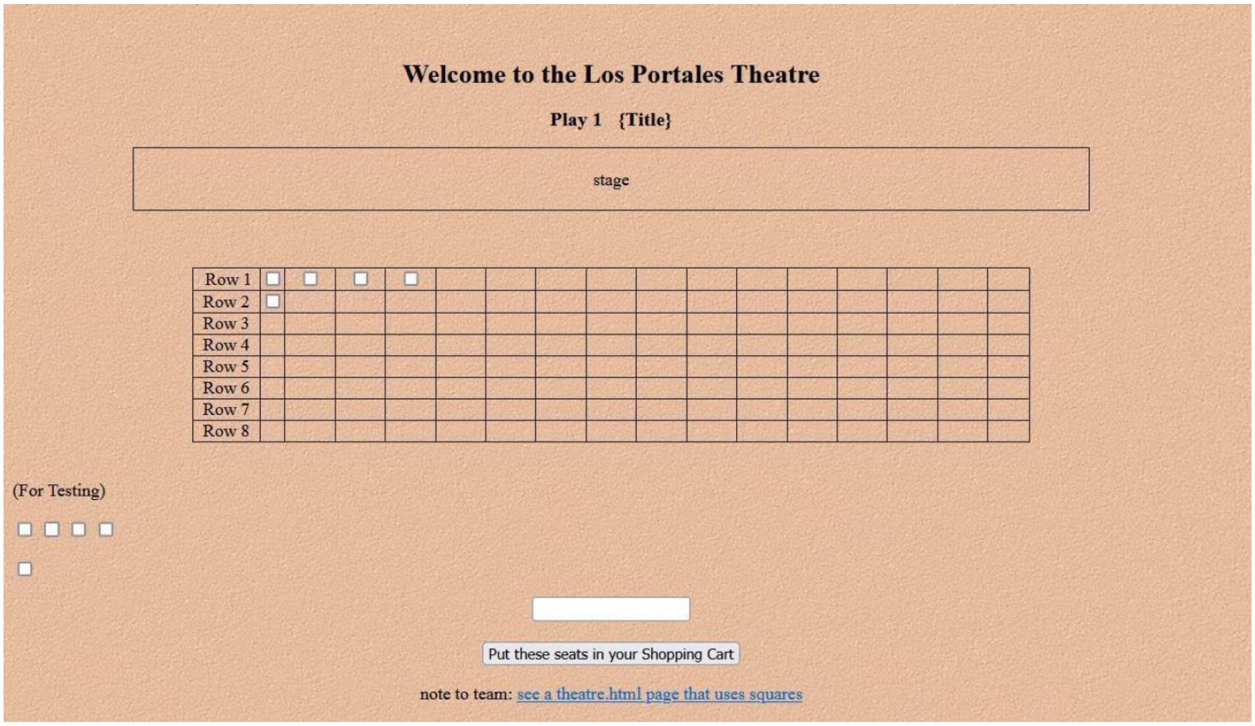
Telephone:

Email:

Age:

- **The Customer enters their registration information.**
- **Shows if customer registration was successful or unsuccessful.**

Figure 14 - Theater Page



Welcome to the Los Portales Theatre

Play 1 {Title}

Row 1	□	□	□	□												
Row 2	□															
Row 3																
Row 4																
Row 5																
Row 6																
Row 7																
Row 8																

(For Testing)

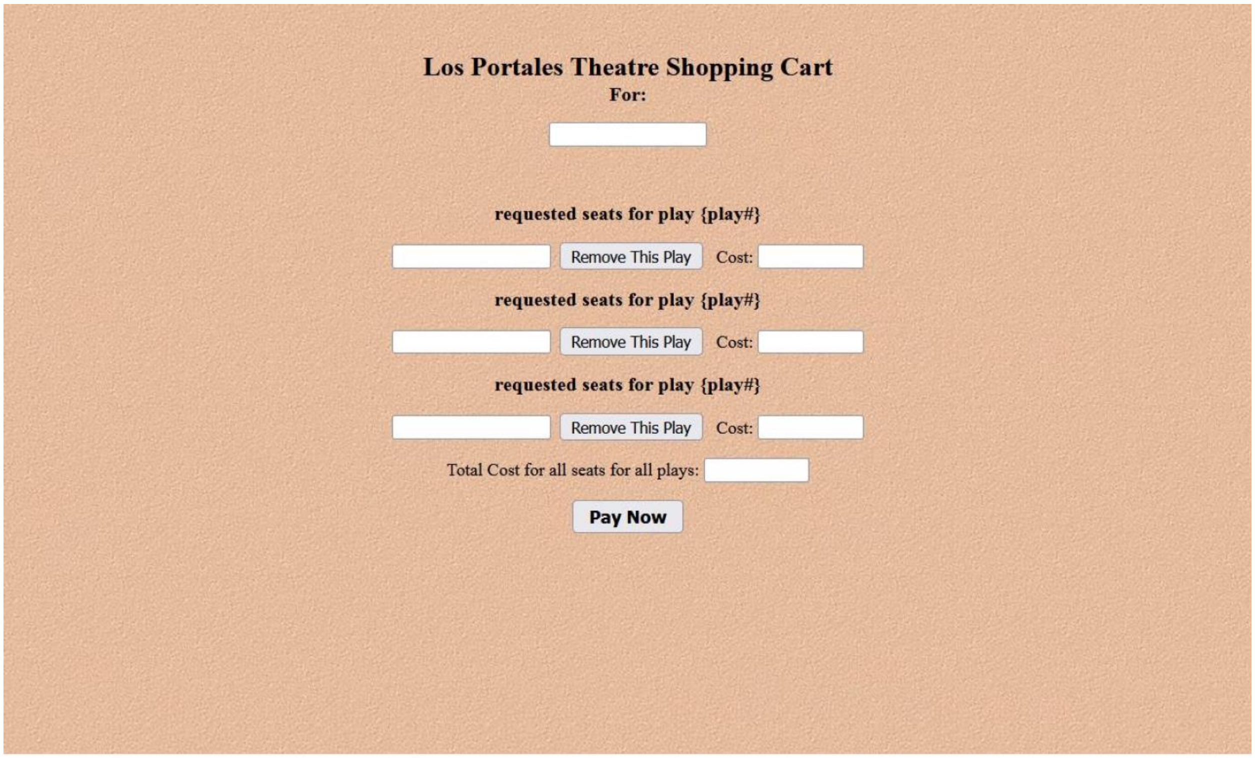
□ □ □ □

□

note to team: see a theatre.html page that uses squares

- **Shows available seats for a play the customer chooses.**
- **The customer can select and unselect seats for that play.**

Figure 15 - Shopping Cart



Los Portales Theatre Shopping Cart

For:

requested seats for play {play#}

Cost:

requested seats for play {play#}

Cost:

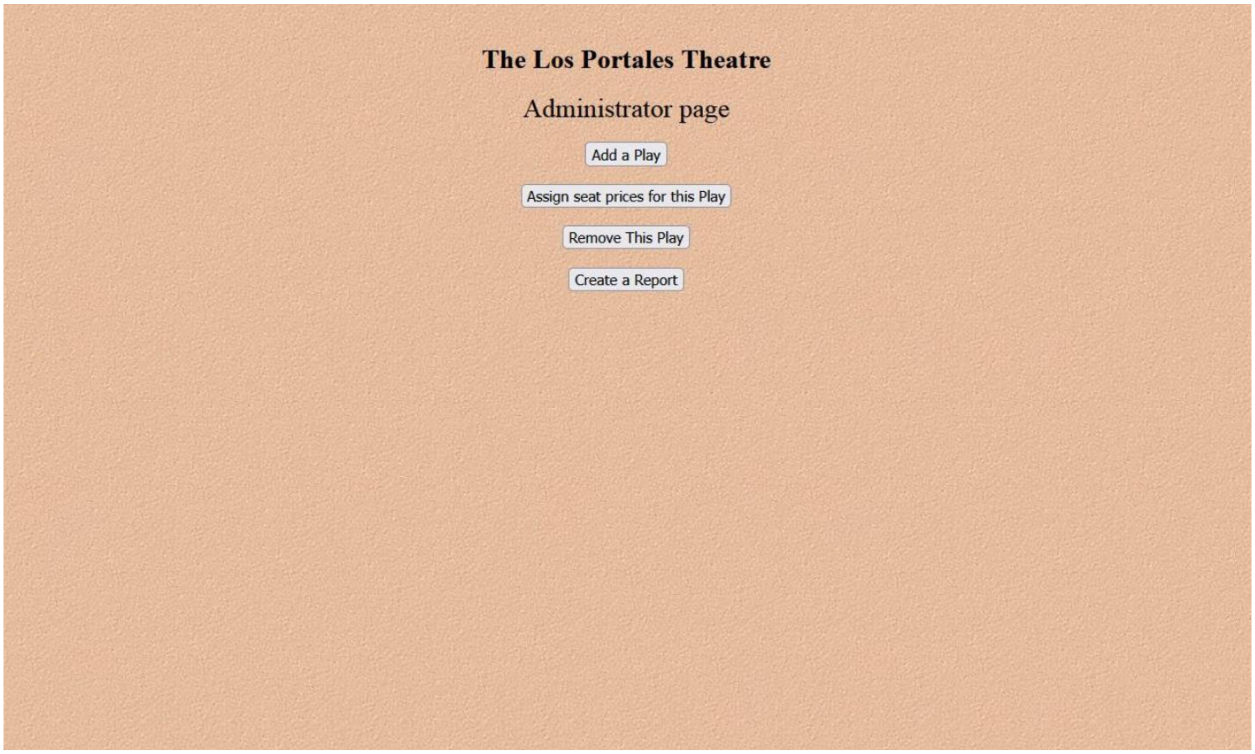
requested seats for play {play#}

Cost:

Total Cost for all seats for all plays:

- **Shows which seats in all shows the customer has purchased and their total cost.**
- **Also on the shopping cart page the customer can unselect seats that they have selected before (and the cost will be adjusted).**
- **The customer enters their payment information.**
- **Shows if the customer payment information entered was not accepted (then the software would show a message.)**

Figure 16 - Administrator Page



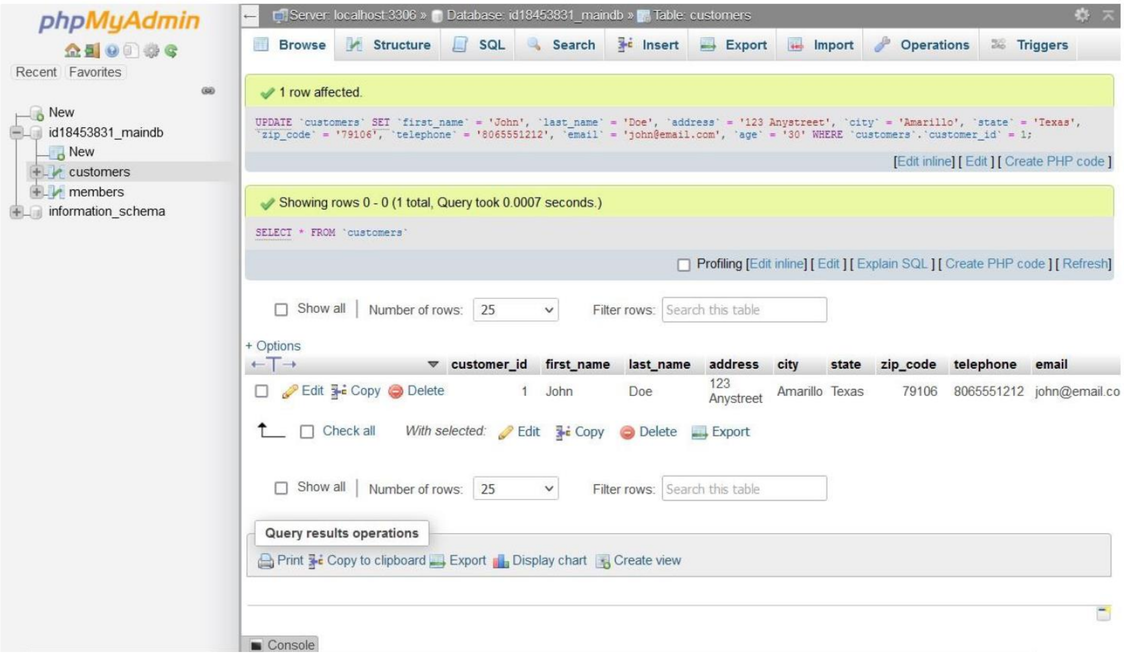
The Los Portales Theatre

Administrator page

- **To add plays with their titles, descriptions, dates and times.**
- **To establish seat pricing for all shows.**
- **To make a system request to generate a report showing how many eats have been sold for a specific play and date.**

Example Customers Table in system DB

Our Customers Table in our database



✓ 1 row affected.

```
UPDATE `customers` SET `first_name` = 'John', `last_name` = 'Doe', `address` = '123 Anystreet', `city` = 'Amarillo', `state` = 'Texas',
`zip_code` = '79106', `telephone` = '8065551212', `email` = 'john@email.com', `age` = '30' WHERE `customers`.`customer_id` = 1;
```

[Edit inline] [Edit] [Create PHP code]

✓ Showing rows 0 - 0 (1 total, Query took 0.0007 seconds.)

```
SELECT * FROM `customers`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table

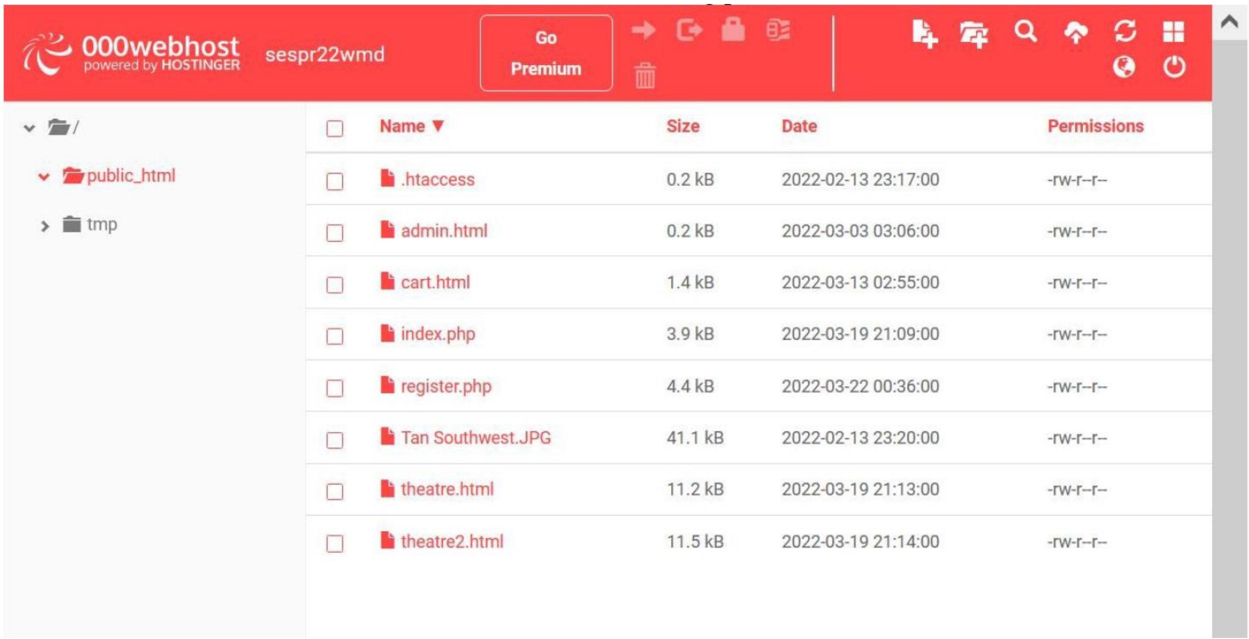
+ Options

	customer_id	first_name	last_name	address	city	state	zip_code	telephone	email
□ Edit Copy Delete	1	John	Doe	123 Anystreet	Amarillo	Texas	79106	8065551212	john@email.co

□ Check all *With selected:* Edit Copy Delete Export

□ Show all | Number of rows: 25 Filter rows: Search this table

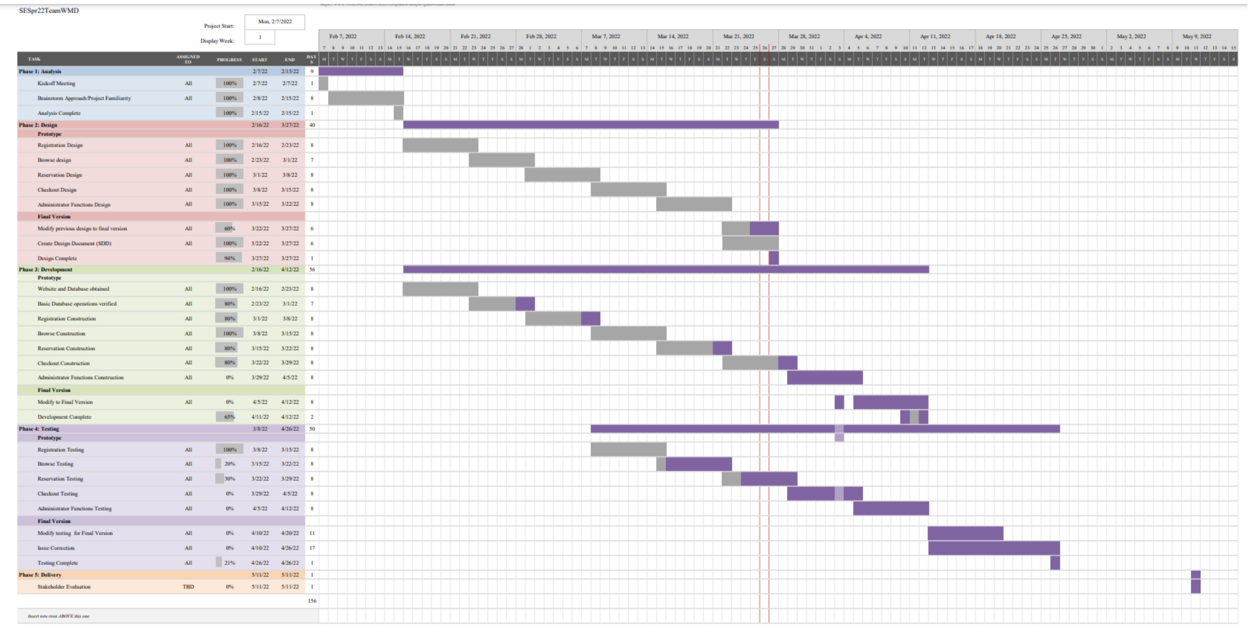
Visual Reference of hosting system files



	Name ▼	Size	Date	Permissions
□	.htaccess	0.2 kB	2022-02-13 23:17:00	-rw-r--r--
□	admin.html	0.2 kB	2022-03-03 03:06:00	-rw-r--r--
□	cart.html	1.4 kB	2022-03-13 02:55:00	-rw-r--r--
□	index.php	3.9 kB	2022-03-19 21:09:00	-rw-r--r--
□	register.php	4.4 kB	2022-03-22 00:36:00	-rw-r--r--
□	Tan Southwest.JPG	41.1 kB	2022-02-13 23:20:00	-rw-r--r--
□	theatre.html	11.2 kB	2022-03-19 21:13:00	-rw-r--r--
□	theatre2.html	11.5 kB	2022-03-19 21:14:00	-rw-r--r--

5.0 Time consideration

5.1 Project Gannt Chart



SESpr22TeamWMD

Project Start: Mon, 2/7/2022

Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END	DAYS
Phase 1: Analysis			2/7/22	2/15/22	9
Kickoff Meeting	All	100%	2/7/22	2/7/22	1
Brainstorm Approach/Project Familiarity	All	100%	2/8/22	2/15/22	8
Analysis Complete		100%	2/15/22	2/15/22	1
Phase 2: Design			2/16/22	3/27/22	40
Prototype					
Registration Design	All	100%	2/16/22	2/23/22	8
Browse design	All	100%	2/23/22	3/1/22	7
Reservation Design	All	100%	3/1/22	3/8/22	8
Checkout Design	All	100%	3/8/22	3/15/22	8
Administrator Functions Design	All	100%	3/15/22	3/22/22	8
Final Version					
Modify previous design to final version	All	60%	3/22/22	3/27/22	6
Create Design Document (SDD)	All	100%	3/22/22	3/27/22	6
Design Complete		94%	3/27/22	3/27/22	1
Phase 3: Development			2/16/22	4/12/22	56
Prototype					
Website and Database obtained	All	100%	2/16/22	2/23/22	8
Basic Database operations verified	All	80%	2/23/22	3/1/22	7
Registration Construction	All	80%	3/1/22	3/8/22	8
Browse Construction	All	100%	3/8/22	3/15/22	8
Reservation Construction	All	80%	3/15/22	3/22/22	8
Checkout Construction	All	80%	3/22/22	3/29/22	8
Administrator Functions Construction	All	0%	3/29/22	4/5/22	8
Final Version					
Modify to Final Version	All	0%	4/5/22	4/12/22	8
Development Complete		65%	4/11/22	4/12/22	2
Phase 4: Testing			3/8/22	4/26/22	50
Prototype					
Registration Testing	All	100%	3/8/22	3/15/22	8
Browse Testing	All	20%	3/15/22	3/22/22	8
Reservation Testing	All	30%	3/22/22	3/29/22	8
Checkout Testing	All	0%	3/29/22	4/5/22	8
Administrator Functions Testing	All	0%	4/5/22	4/12/22	8
Final Version					
Modify testing for Final Version	All	0%	4/10/22	4/20/22	11
Issue Correction	All	0%	4/10/22	4/26/22	17
Testing Complete	All	21%	4/26/22	4/26/22	1
Phase 5: Delivery			5/11/22	5/11/22	1
Stakeholder Evaluation	TBD	0%	5/11/22	5/11/22	1
					156

Insert new rows ABOVE this one

Figure 17 - The prototype subheadings in this Gantt chart's phase 2 and 3 sections refers to a limited version of the software that uses only one customer and only one play. The reasoning for making the prototype is that if such a prototype cannot be accomplished then obviously a full version of the software with many customers and many plays cannot be accomplished. Fortunately, however, construction of such a prototype is estimated to be around 90% of the construction of the final version, so once the prototype is made, there would be just a few modifications needed.

5.2 Project Milestones

All dates are final dates for items to be completed.

1. Requirements Specification – February 27th
    a. Create Pert, CPM, Gantt, Cost Diagram – February 27th
    b. Requirement Description Document – February 27th
2. Software requirements specification – February 27th
    a. Create use case, class, and Sequence diagrams – February 27th
3. Design Specification – March 27th
    a. Update Gantt and other diagrams – March 27th
    b. Limited version of software created – March 27th
4. Risk Analysis and Test specification – April 17th
    a. Software test plan – April 17th

b. Final version of software – April 17th

c. Testing for bugs and issues – April 17th

5. Presentation to Customer and stockholders – May 11th