Software Requirements Specification

Version 2.0
<<Stage 2>>

March 27, 2022

Theatre Booking System

Stephen Villanueva
Trey Allen
Giuseppe Scalise
Robert Leal

CHANGELOG		
Version	**Description**	**Member**
1.0	Initial documentation	All
2.0	• Added use case tables, security features, and functional / non-functional requirements. • Modified use-case table hierarchy (*System Environmen*t). • Adjusted diagrams. • Highlighted changes made in this version.	Trey

Table of Contents

List of Figures

1.0. Introduction

1.1. Purpose

The purpose of this document is to present a detailed description of the Theatre Booking System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the Los Portales Theatre for its approval.

1.2. Scope of Project

This software system will be a Theatre Booking System for a local theatre that hosts plays at their facility. This system will be designed to increase business continuity efficiency by providing an interactive menu that allows customers and staff to automate the ticket booking process which is normally performed and analyzed/modified manually. The system will meet the requirements of the facility while being easily accessible on the internet and user-friendly for customers – and saving paper costs by transacting online.

More specifically, this system is designed to allow an administrator (owner, staff members, and more) to manage play showtimes and prices, and generate reports regarding the ticket sells for a certain play during a specific showtime. Furthermore, this system will allow registered customers (viewers) to add and remove seats for any given play, buy seats that have been added, and receive a detailed report of their purchase. The software communicates via email, online servers, and the configuration set by the administrator from the maintenance module provided on the software's interface.

1.3. Glossary

Term	Definition
Administrator	A person responsible for running the system software.
Business Continuity	Plan to deal with difficult situations; continuing to function with as little disruption as possible.
Customer Module	Key activities for the registered customer to add and remove seats, add and remove to cart, change from different plays, and purchase tickets.
Database	Collection of all the information monitored by this system.
Developers	A person that is responsible for creating the (working) system.
Field	A cell within a form.
Graphical User Interface	A visual way of interacting with a computer.
Maintenance Module	Key activities for the administrator to create, manage, and delete fields.
Registered Customer	Basic user information stored in the database that grants the user access to Customer Module
Report	Database object that displays or distributes a summary of data.
Software Requirements Specification	A document that completely describes all the functions of a proposed system and the constraints under which it must operate. For example, this document.
Stakeholder	Any person with an interest in the project who is not a developer.
User	Customer or Administrator

1.4. References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

1.5. Overview of Document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety but are intended for different audiences and thus use different language.

2.0. Overall Description

2.1 System Environment

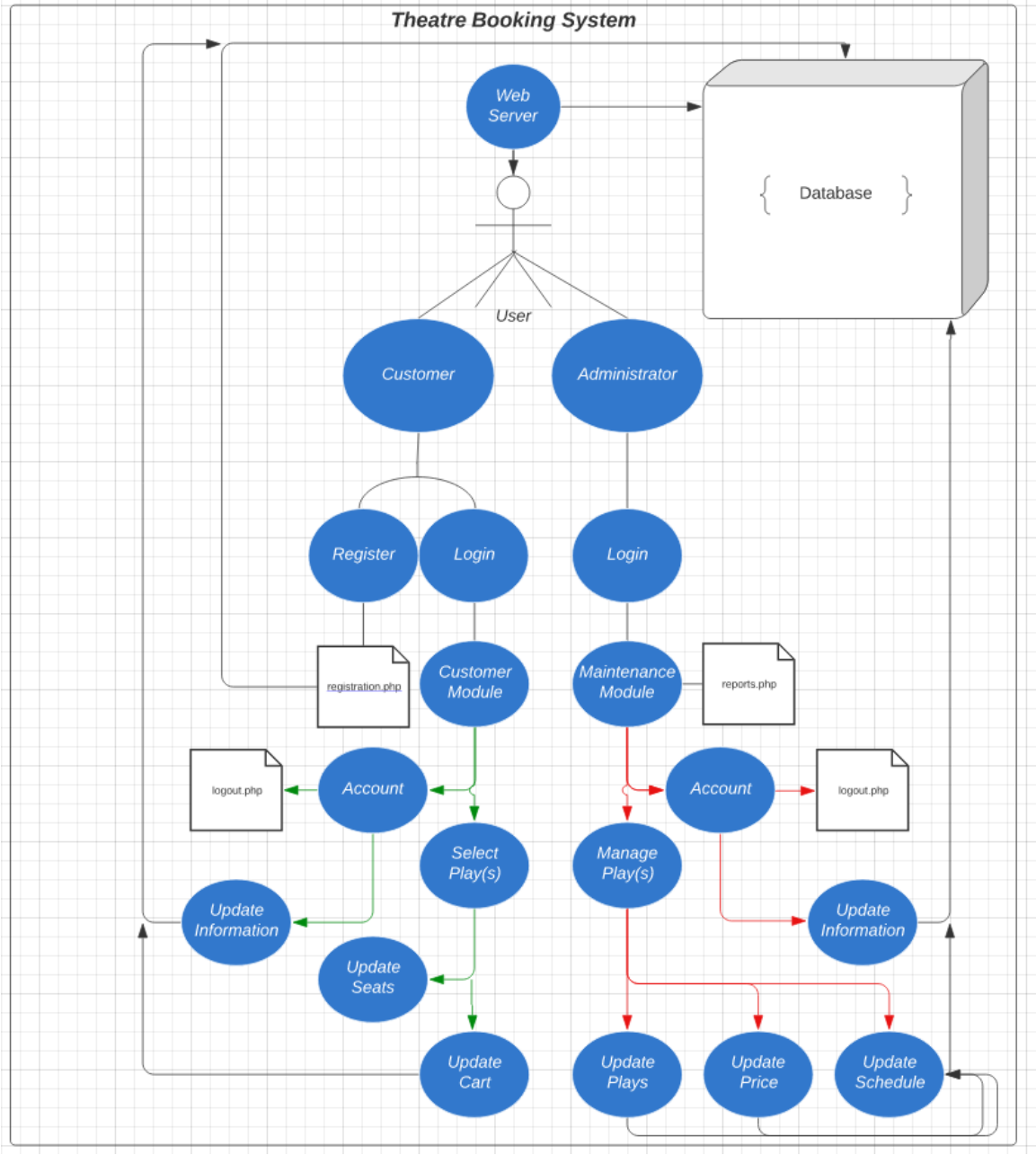


Figure 1 - System Environment

The Theatre Booking System has a single actor who is prompted with an interactive logon screen. The single actor will provide their credentials through the web application and be granted privileges accordingly. The graphical user interface will allow the actor to communicate with the system software and integrate with the backend database that stores information about the plays, scheduling, purchasing, and registered customers/administrators. The database will speak to an email server for reporting.

2.2 Functional Requirements Specification

This section outlines the use-cases for the single actor in both the customer and administrator perspective.

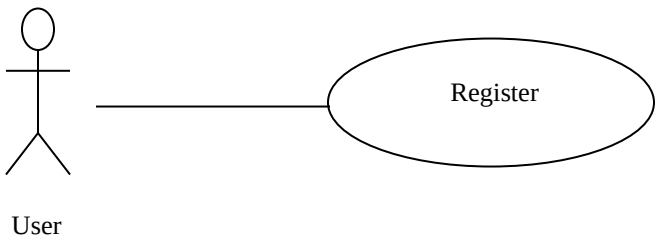
2.2.1 User Use Case

Use case: **Register**

Table:

Use-case ID:	user_register		
Use-case Name:	Register		
Created By:	Trey A.	Last Updated By:	
Date Created:	3/27/22	Date Last Updated:	
Actor:	User - likely subclass Customer as Admin account is set by developers		
Description:	Allows one to interact with the Point of Sale system		
Preconditions:	Internet connection, device to navigate to web page, peripherals to register		
Post conditions:	Added to MySQL database; must remember email:password combination		
Priority:	High		
Frequency of Use:	TBD		
Normal Course of Events:	After exploring landing page, a user will be presented with a fillable form to proceed in viewing and purchasing play tickets.		
Alternative Courses:	User can enter an incorrect email:password combination, lose Internet connection, or tries to register with an email address already used.		
Exceptions:	Administrators		
Includes:	All		
Special Requirements:			
Assumptions:			
Notes and Issues:			

Diagram:



Brief Description
The User's only use case to access system functionality is to register to the database.

Initial Step-By-Step Description
Before this use case can be initiated, the User must access the web application software.

1. The User chooses the Register button.
2. The system displays fillable fields to the User.
3. The User provides their name, age, address, telephone number, and email address.
4. The system confirms the registration was successful.
5. The User chooses the Login button.
6. The system displays the corresponding Module depending on credentials.

2.2.2 Registered Customer Use Case

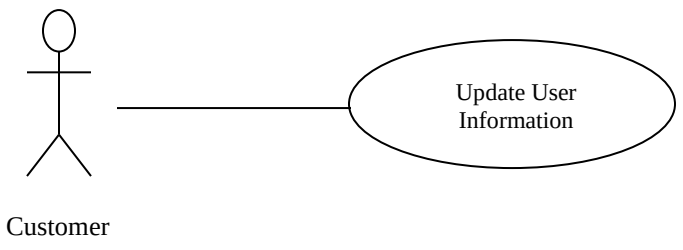
Depending on credentials of the user, this case covers a customer's role-based access.

Use case: Update User Information

Table:

Use-case ID:	update_customer_information		
Use-case Name:	Update Information		
Created By:	Trey A.	Last Updated By:	
Date Created:	3/27/22	Date Last Updated:	
Actor:	Customer		
Description:	Allows Customer to edit account details (phone number, address, etc.)		
Preconditions:	Use case: Register		
Post conditions:	Tables in MySQL database are edited		
Priority:	Low		
Frequency of Use:	Low		
Normal Course of Events:	After authenticating to the web application, the Customer can navigate to the account page and edit their details.		
Alternative Courses:	User cannot visit account page due to invalid session		
Exceptions:	Users		
Includes:	All		
Special Requirements:			
Assumptions:			
Notes and Issues:			

Diagram:



Brief Description
The registered customer can edit the data submitted during the registration process.

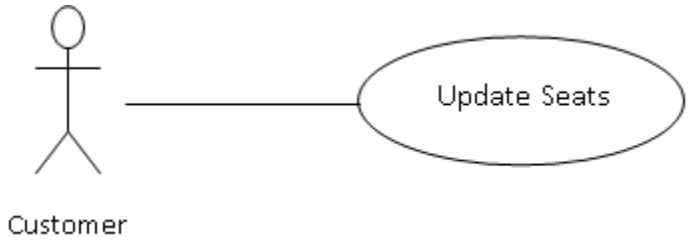
Initial Step-By-Step Description
Before this use case can be initiated, the User must access the web application software and login using the credentials created in the Register (2.2.1) use case.

1. The Registered Customer chooses the *Edit Profile* button.
2. The system presents the registered customer with the registration fields.
3. The Registered Customer will change the fields as needed and select *Save.*
4. The System will update the database.

Use case: Update Seats
Table:

Use-case ID:	update_seats		
Use-case Name:	Update Seats		
Created By:	Trey A.	Last Updated By:	
Date Created:	3/27/22	Date Last Updated:	
Actor:	Customer		
Description:	Allows Customer to select and deselect their seating arrangement for the play		
Preconditions:	Use case: Register		
Post conditions:	Server using PHP sessions to cache current selection		
Priority:	High		
Frequency of Use:	High		
Normal Course of Events:	After authenticating to the web application, the Customer can navigate to the various play times and select seats based on availability		
Alternative Courses:	User is not authenticated, seat is taken (not clickable), or connection is lost		
Exceptions:	Administrators		
Includes:	All		
Special Requirements:			
Assumptions:			
Notes and Issues:			

Diagram:



Brief Description
The registered customer can their currently selected seats (add/remove).

Initial Step-By-Step Description
Before this use case can be initiated, the User must access the web application software and login using the credentials created in the Register (2.2.1) use case.

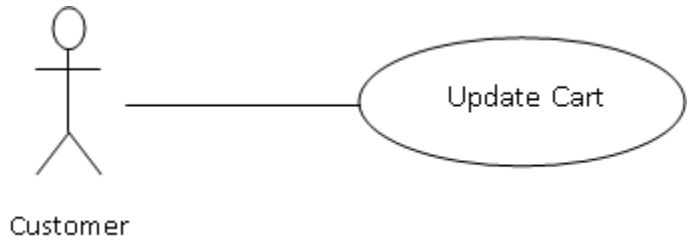
1. The system will present a list of available seats in green for a specific showing.
2. The Registered Customer can select an available seat for any available showing.
3. If the Registered Customer needs a seat from another play, the RC will select *Plays.*
4. The system will present all showings configured by the administrator.
5. The Registered Customer will repeat steps 1-3 until they have fulfilled their desire.

Use case: Update Cart

Table:

Use-case ID:	update_cart		
Use-case Name:	Update Cart		
Created By:	Trey A.	Last Updated By:	
Date Created:	3/27/22	Date Last Updated:	
Actor:	Customer		
Description:	Allows Customer to verify items are correct before purchasing		
Preconditions:	Use case: Register, Use case: Update Seats		
Post conditions:	Sent to payment processor and generates receipt		
Priority:	High		
Frequency of Use:	Medium		
Normal Course of Events:	After authenticating to the web application, the Customer can navigate to the various play times and select seats based on availability, update seats, then prepare to purchase		
Alternative Courses:	Customer can leave the cart and not proceed with transaction		
Exceptions:	Administrators		
Includes:	All		
Special Requirements:			
Assumptions:			
Notes and Issues:			

Diagram:



Brief Description

The registered customer can add or remove their selected seats (Update Seats use case) to their shopping cart to prepare for a purchase.

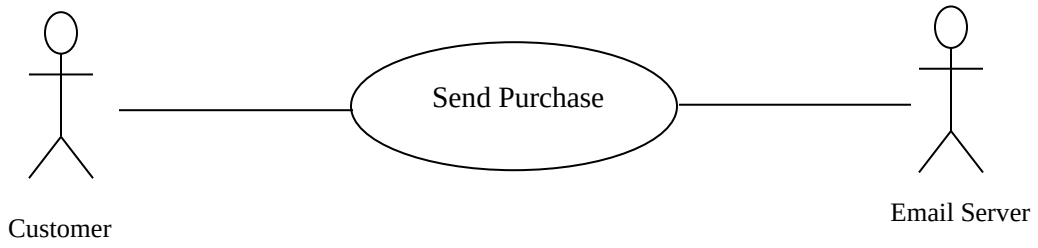
Initial Step-By-Step Description
Before this use case can be initiated, the User must access the web application software and login using the credentials created in the Register (2.2.1) use case. Then, have seats selected from the Update Seat use case.

1. The Registered Customer chooses the *Add to Cart* button.
2. The system will store their selected seats into a database.
3. The Registered Customer chooses *View Cart* to see tickets that area ready for purchase.
4. The system will display icons for adding or removing items to the cart and update DB.

Use case: Send Purchase
Table:

Use-case ID:	complete_transaction		
Use-case Name:	Purchase		
Created By:	Trey A.	Last Updated By:	
Date Created:	3/27/22	Date Last Updated:	
Actor:	Customer		
Description:	After verifying cart, this will secure tickets requested by Customer; pending payment		
Preconditions:	Use case: Register, Use case: Update Seats, Use case: Update Cart		
Post conditions:	Receipt generated; MySQL database will remove purchased seats from availability		
Priority:	High		
Frequency of Use:	Medium		
Normal Course of Events:	Cart will be updated and payment information will be submitted into form. Upon successful payment processing, users will be redirected to success page.		
Alternative Courses:	Payment fails or returns to seat selection		
Exceptions:			
Includes:	All		
Special Requirements:			
Assumptions:			
Notes and Issues:			

Diagram:



Brief Description
The user can submit their purchase after providing credit card information to the database. The registered costumer will receive an email confirmation of their order.

Initial Step-By-Step Description
Before this use case can be initiated, the User must access the web application software and login using the credentials created in the Register (2.2.1) use case. Then, have seats selected from the Update Seat use case and have items in their cart (Update Cart).

1. The Registered Customer chooses the *View Cart* button.
2. The system will display all tickets added to the RC's cart and display a *Checkout* option.
3. The Registered Customer chooses *Checkout.*
4. The system will display fields for credit card information and a *Submit* button.
5. The system will display a success or error message and display the output accordingly.

2.2.3 Administrator Use Cases
The Administrator has the following sets of use cases:

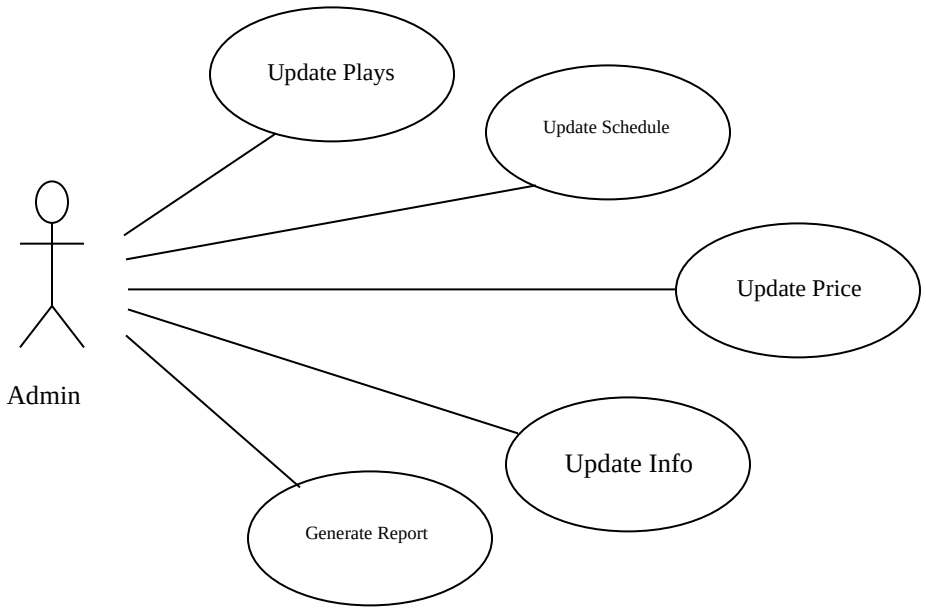


Figure 2 - Administrator Use Cases

Use case: Update Information

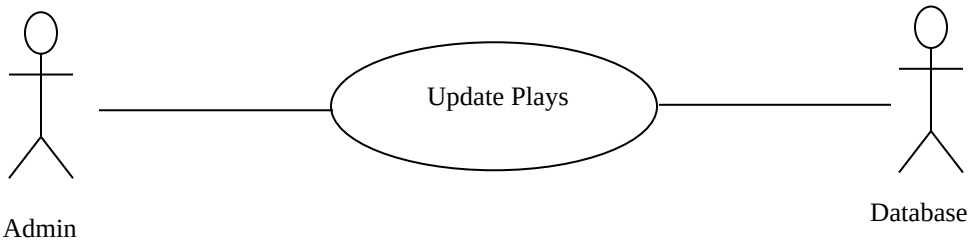
Xref: Section 2.2.2, Update User Information

Note
This use case will work in the same manner as referenced above. The different modules (Customer and Maintenance) will not affect this functionality.

Use case: Update Plays
Table:

Use-case ID:	update_plays		
Use-case Name:	Update Plays		
Created By:	Trey A.	Last Updated By:	
Date Created:	3/27/22	Date Last Updated:	
Actor:	Administrator		
Description:	This will allow the creation, updating, or removal of future showings		
Preconditions:	Use case: Login -> Administrator		
Post conditions:	MySQL update -> Customers see change in the Customer Module		
Priority:	High		
Frequency of Use:	High		
Normal Course of Events:	Administrator will authenticate and select "Update Plays" from the list of options in the Maintenance Module		
Alternative Courses:	Authentication fails; Enters different section		
Exceptions:			
Includes:	Administrators		
Special Requirements:	Privileged access		
Assumptions:			
Notes and Issues:	If Customers have already purchased a play that gets deleted, refunds need to be sent		

Diagram:



Brief Description
The administrator creates, updates, or removes a play/showing.

Initial Step-By-Step Description
Before this can be initiated, the administrator must access the web application software and authenticate accordingly. Then, the administrator must navigate to the Play section of the Maintenance module.

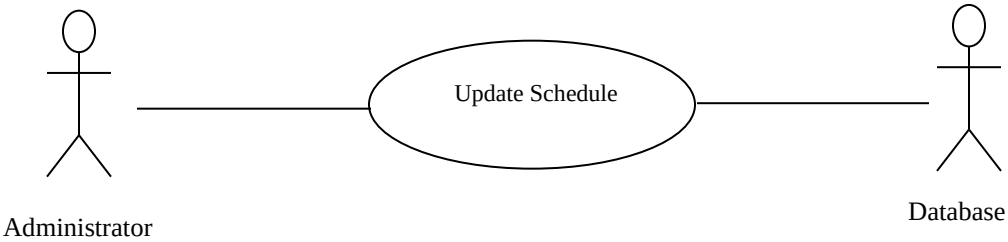
1. The Administrator selects *Plays* from the Maintenance module.
2. The system presents a choice of adding, updating, or removing.
3. The Administrator chooses to add, update, or remove.
4. The system links to the existing database.
5. Depending on the choice, administrator will be prompted according to Figure 4.
6. The Editor fills in the information and submits the form.
7. The system verifies the information and updates the database.

Use case: Update Schedule

Table:

Use-case ID:	update_schedule		
Use-case Name:	Update Schedule		
Created By:	Trey A.	Last Updated By:	
Date Created:	3/27/22	Date Last Updated:	
Actor:	Administrator		
Description:	This will allow editing of the times of existing plays		
Preconditions:	Use case: Login -> Administrator		
Post conditions:	MySQL update -> Customers see change in the Customer Module		
Priority:	High		
Frequency of Use:	Low		
Normal Course of Events:	Administrator will authenticate and select "Update Schedule" from the list of options in the Maintenance Module		
Alternative Courses:	Authentication fails; Enters different section		
Exceptions:			
Includes:	Administrators		
Special Requirements:	Privileged access		
Assumptions:			
Notes and Issues:	Email should be sent out to paying Customers notifying them of the time change		

Diagram:



Brief Description

The administrator can manage the scheduling of plays.

Initial Step-By-Step Description
Before this can be initiated, the administrator must access the web application software and authenticate accordingly. Then, the administrator must navigate to the Schedule section of the Maintenance module.

1. The Administrator selects *Schedule* from the Maintenance module.
2. The system presents a list of current Plays.
3. The Administrator chooses a play to schedule (or reschedule).
4. The system links to the existing database.
5. The system provides fields relevant to the Play's showtimes.
6. The Administrator fills in the information and submits the form.
7. The system verifies the information and updates the database.

Use case: Update Price

Table:

Use-case ID:	update_price		
Use-case Name:	Update Price		
Created By:	Trey A.	Last Updated By:	
Date Created:	3/27/22	Date Last Updated:	
Actor:	Administrator		
Description:	This will allow editing of the prices for a specific seat in the theatre		
Preconditions:	Use case: Login -> Administrator		
Post conditions:	MySQL update -> Customers see change in the Customer Module		
Priority:	High		
Frequency of Use:	Low		
Normal Course of Events:	Administrator will authenticate and select "Update Price" from the list of options in the Maintenance Module		
Alternative Courses:	Authentication fails; Enters different section		
Exceptions:			
Includes:	Administrators		
Special Requirements:	Privileged access		
Assumptions:			
Notes and Issues:	Price should not be editable if the seat has already been reserved		

Diagram:

Brief Description

The administrator can manage the price of the tickets.

Initial Step-By-Step Description
Before this can be initiated, the administrator must access the web application software and authenticate accordingly. Then, the administrator must navigate to the Price section of the Maintenance module.

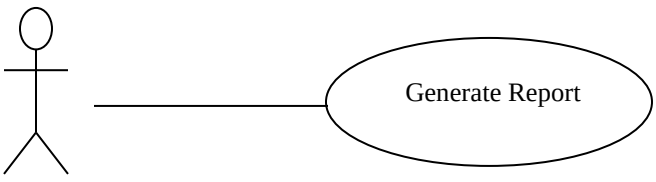
1. The Administrator selects *Price* from the Maintenance module.
2. The system presents a list of current Plays.
3. The Administrator chooses a play to manage seat prices.
4. The system links to the existing database.
5. The system provides fields relevant to the pricing of tickets.
6. The Administrator fills in the information and submits the form.
7. The system verifies the information and updates the database.

Use case: Generate Report

Table:

Use-case ID:	generate_report		
Use-case Name:	Generate Report		
Created By:	Trey A.	Last Updated By:	
Date Created:	3/27/22	Date Last Updated:	
Actor:	Administrator		
Description:	This will allow a the admin to generate a detailed report about ticket sale information		
Preconditions:	Use case: Login -> Administrator		
Post conditions:	MySQL retrieve -> Report will display on screen and send an email		
Priority:	High		
Frequency of Use:	Medium		
Normal Course of Events:	Administrator will authenticate and select "Generate Report" from the list of options in the Maintenance Module, then specify a play.		
Alternative Courses:	Authentication fails; Enters different section		
Exceptions:			
Includes:	Administrators		
Special Requirements:	Privileged access		
Assumptions:			
Notes and Issues:			

Diagram:



Administrator

Brief Description

The administrator can generate a report of ticket sales.

Initial Step-By-Step Description
Before this can be initiated, the administrator must access the web application software and authenticate accordingly. Then, the administrator must navigate to the Reports section of the Maintenance module.

1. The Administrator selects *Reports* from the Maintenance module.
2. The system presents a list of Plays.
3. The Administrator chooses a play to see statistics on its sales.
4. The system links to the existing database.
5. The system generates a PDF report.
6. The system emails the report to the administrator.

2.3 User Characteristics

The user is expected to be literate in navigating a GUI.

The Administrator is expected to be literate in navigating a GUI and have familiarity with reading reports in .csv and .pdf formats. Furthermore, beginner knowledge of databases is necessary as this server will be hosted at the facility.

2.4 Non-Functional Requirements

The Theatre Booking System will run on a server provided by contracting staff members. The server is expected to run on a high-speed internet connection along with the database. The workstation and/or tablet to be used by the Administrator and Customers will be provided by Los Portales Theatre. The workstation is expected to have internet capability. Workstation is bilanguage friendly - languages supported: English, Spanish. GUI will translate instantaneously with a click of a button.

The system will operate in a user-friendly, non-burdensome, and optimized interaction (user guide to be provided). The GUI will work seamlessly for any device, including personal mobile phones or tablets (menus adjust to scale). Furthermore, updates and requests will run smoothly and efficiently to avoid disruption and/or User frustration. Administrators will have database access in a secure manner. Refer to 3.3.2 (Security).

3.0. Requirements Specification

3.1 External Interface Requirements

This system software will require a third-party vendor for processing payments. Robert Leal's contracting group does not provide a credit card processing system; however, the system software is able to integrate with any processing software.

The *Credit Card Processing System* will communicate with a server using the Hypertext Transfer Protocol Secure (HTTPS) protocol. All transactions will be secure.

3.2 Functional Requirements

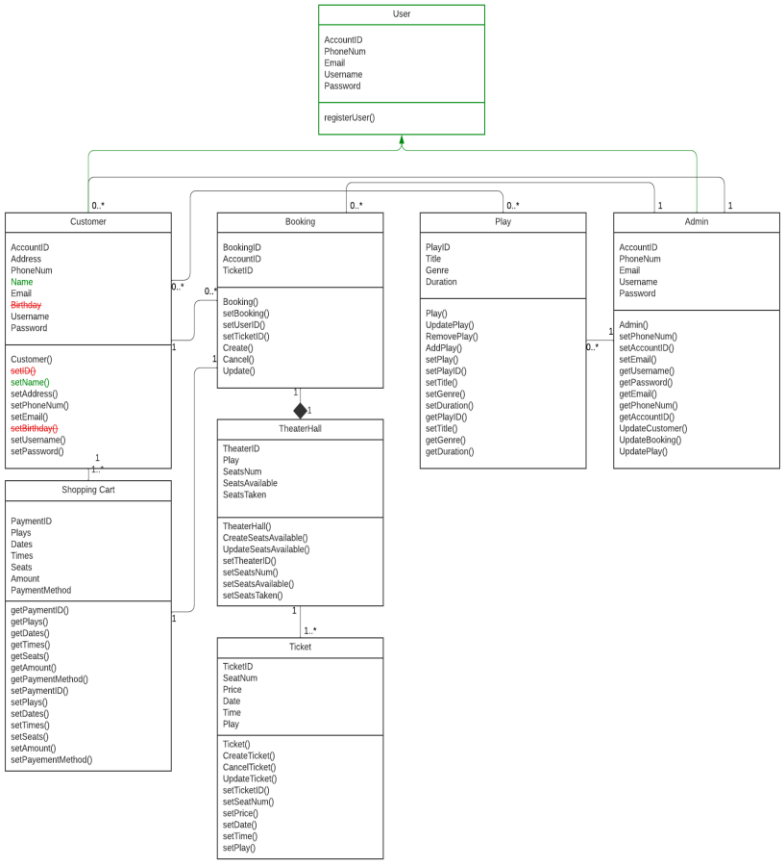


Figure 3 – Class Diagram

3.3 Detailed Non-Functional Requirements

3.3.1 Logical Structure of the Data

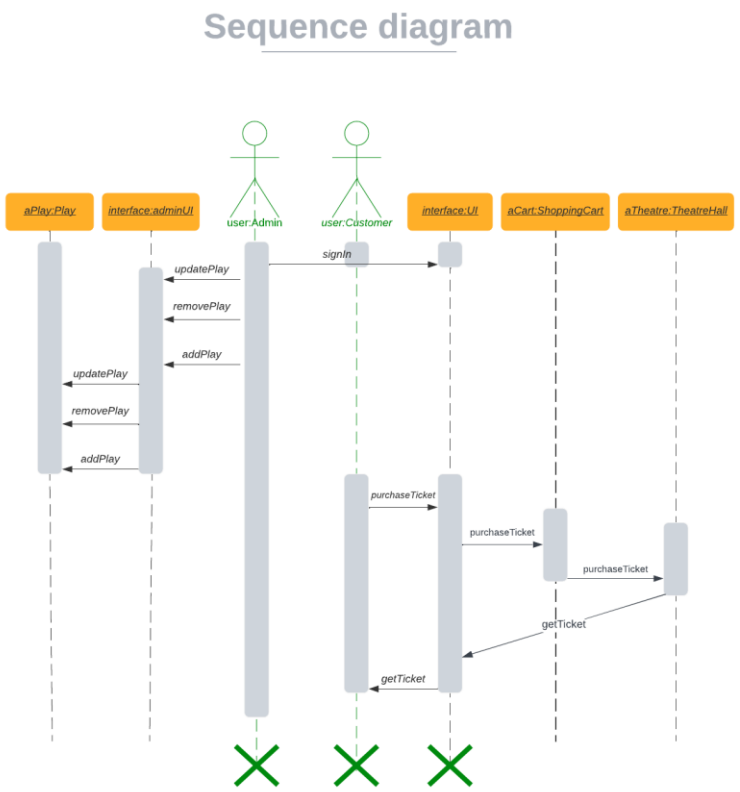


Figure 2 – Sequence Diagram

3.3.2 Security

As addressed above, the server will communicate via HTTPS protocol for secure transmission of data. Furthermore, the server's DNS will route through Cloudflare, a leading provider that ensures traffic is protected from Distributed Denial of Service (DDoS) attacks. The forms' input will be sanitized to prevent code from executing.

The system will run auto-lockout scripts, disable the cache, provide pattern-hiding displays for passwords, only provide the last four digits of a credit card number, and encrypt the databases that are only accessible to administrators of the system software.

As the system software will allow credit card retention with the consent of the Customer, the system will abide by Amazon Web Services (AWS) Service License Agreement (SLA) and safely store information on a private, dedicated host node. These nodes are backed up regularly (once an hour) and are run in server farms for redundancy. Uptime remains 99.9% as of March 27th, 2022. Plenty of storage is provided but can be expanded at the request of the client to ensure business continuity (for extra cost).

The system requires Users (of both subclasses) to create a password that meets minimum password complexity requirements: [passwords stored as BCrypt Hash]

- Minimum of 8 characters
- Use of a capital letter
- Use of a lowercase letter
- Use of a number
- Use of a special character/symbol

The system will accept payment in the form of cryptocurrency secured by the blockchain. Cryptocurrency allows for confidentiality and the inability to obtain transaction information by an unintended party. Third-party payment processors will be responsible for secure transactions, other than the SSL encryption mentioned above.

Product testing to take place to ensure system works efficiently. For example, offering only one theatre room at a single given time to avoid confusion, or attempting to break the system software by simulating peak-hour traffic or malicious users. If the system software can withstand these “penetration” tests, the prototype will be ready to publish for the clients’ viewing.