Software Requirements Specification

Version 1.0
<<Stage 1>>

February 27, 2022

Theatre Booking System

Robert Leal
Trey Allen
Giuseppe Scalise
Stephen Villanueva

Submitted in partial fulfillment
of the requirements of
CS 472 Software Engineering

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	Reviewer or Author.

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

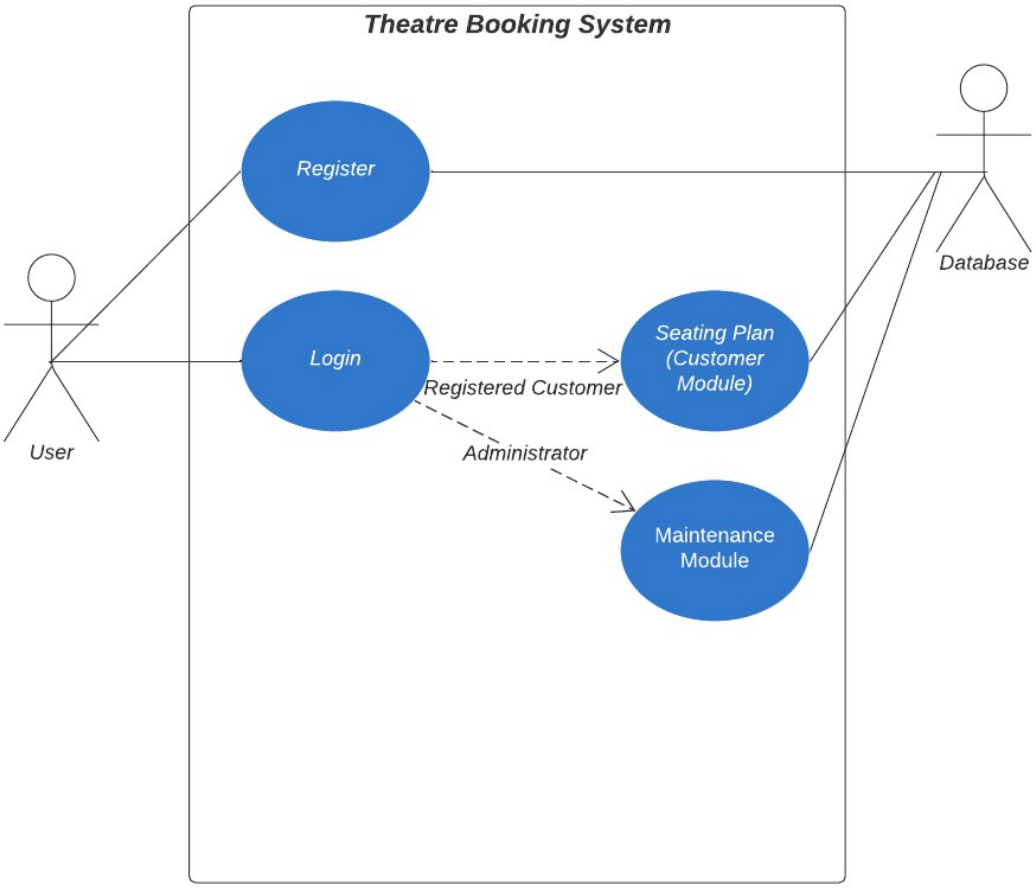


Figure 1 - System Environment

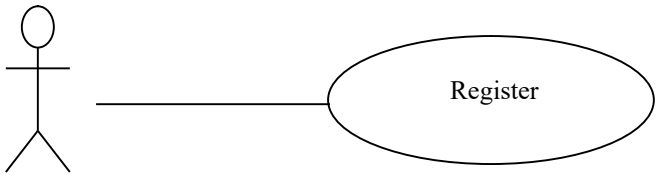
The Theatre Booking System has a single actor who is prompted with an interactive logon screen. The single actor will provide their credentials through the web application and be granted privileges accordingly. The graphical user interface will allow the actor to communicate with the system software 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 registered customer and administrator perspective. *Version 2.0 of this document expects to depict use-cases in detail including pre- and post-conditions, priority, and assumptions.*

2.2.1 User Use Case

Use case: **Register**
Diagram:



User

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

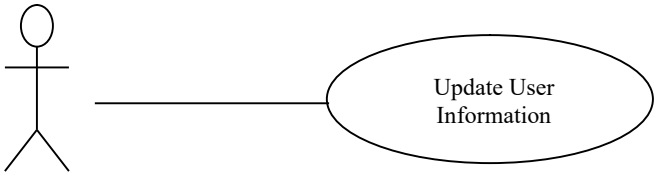
Initial Step-By-Step Description
Before this use case can be initiated, the User must access the web application software.

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
Diagram:



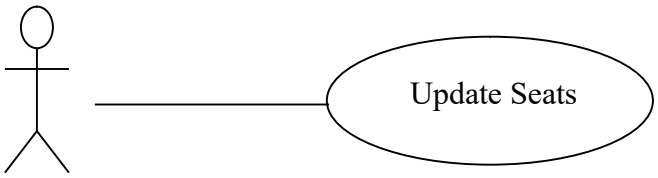
R Customer

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
Diagram:



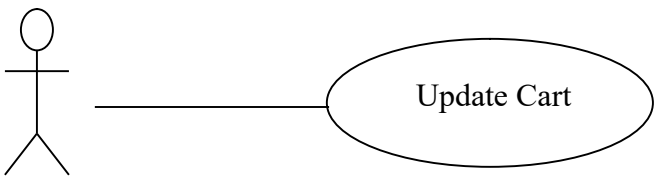
R Customer

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
Diagram:



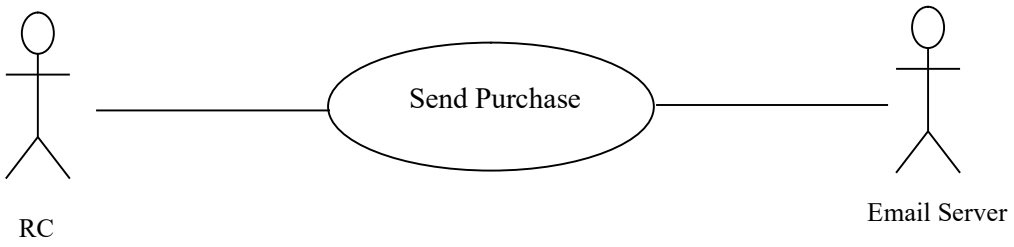
R Customer

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
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:

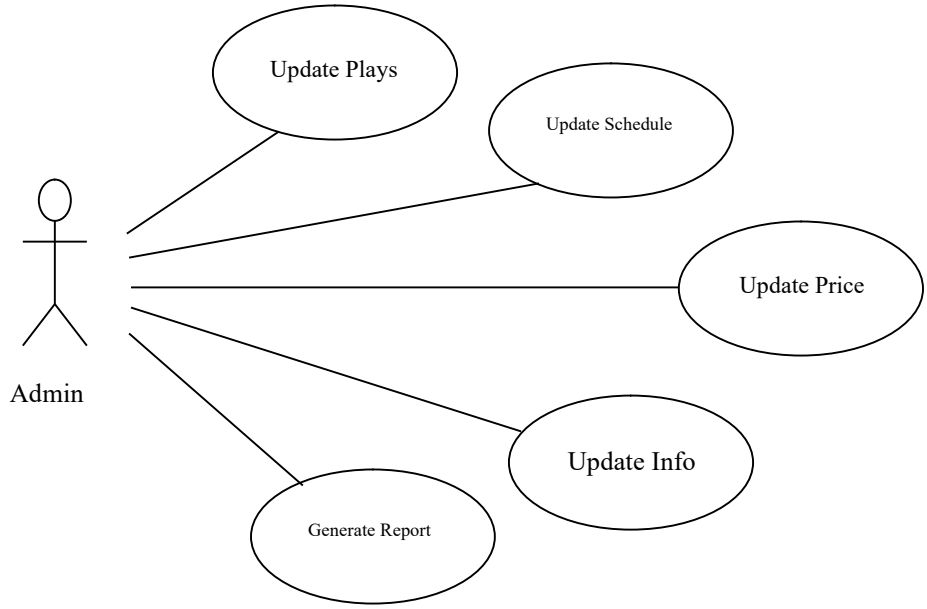


Figure 2 - Administrator Use Cases

Use case: Update Information

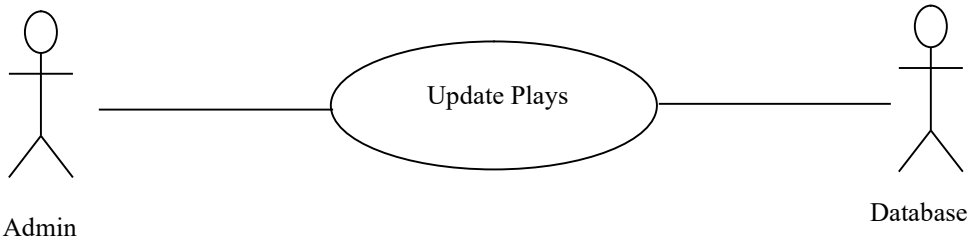
Xref: Section 2.2.2, Update User Information

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

Diagram:



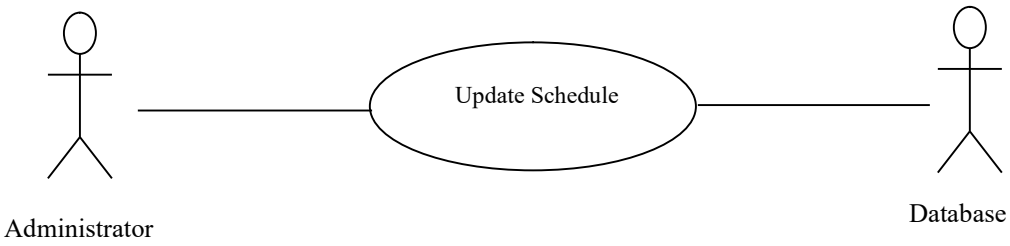
Brief Description

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
Diagram:

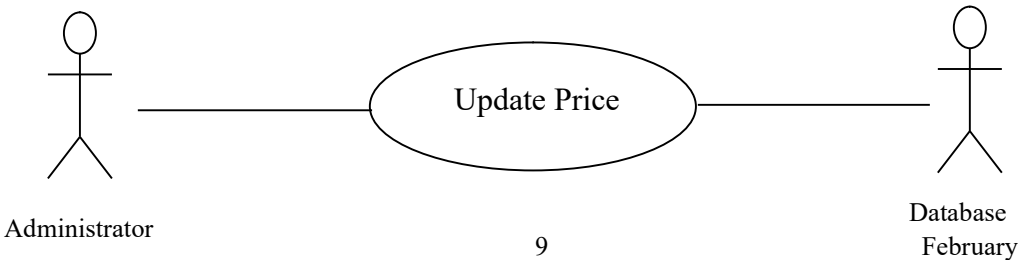


Brief Description
The administrator can manage the scheduling of plays.

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
Diagram:

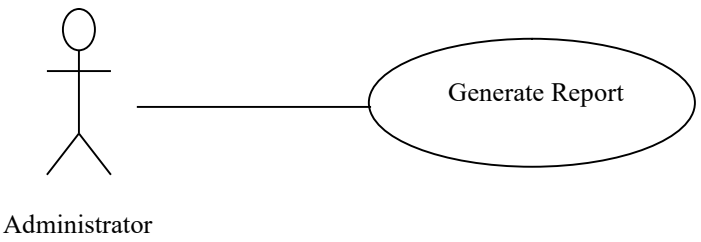


Brief Description
The administrator can manage the price of the tickets.

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
Diagram:



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 Robert Leal's contracting staff. 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.

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

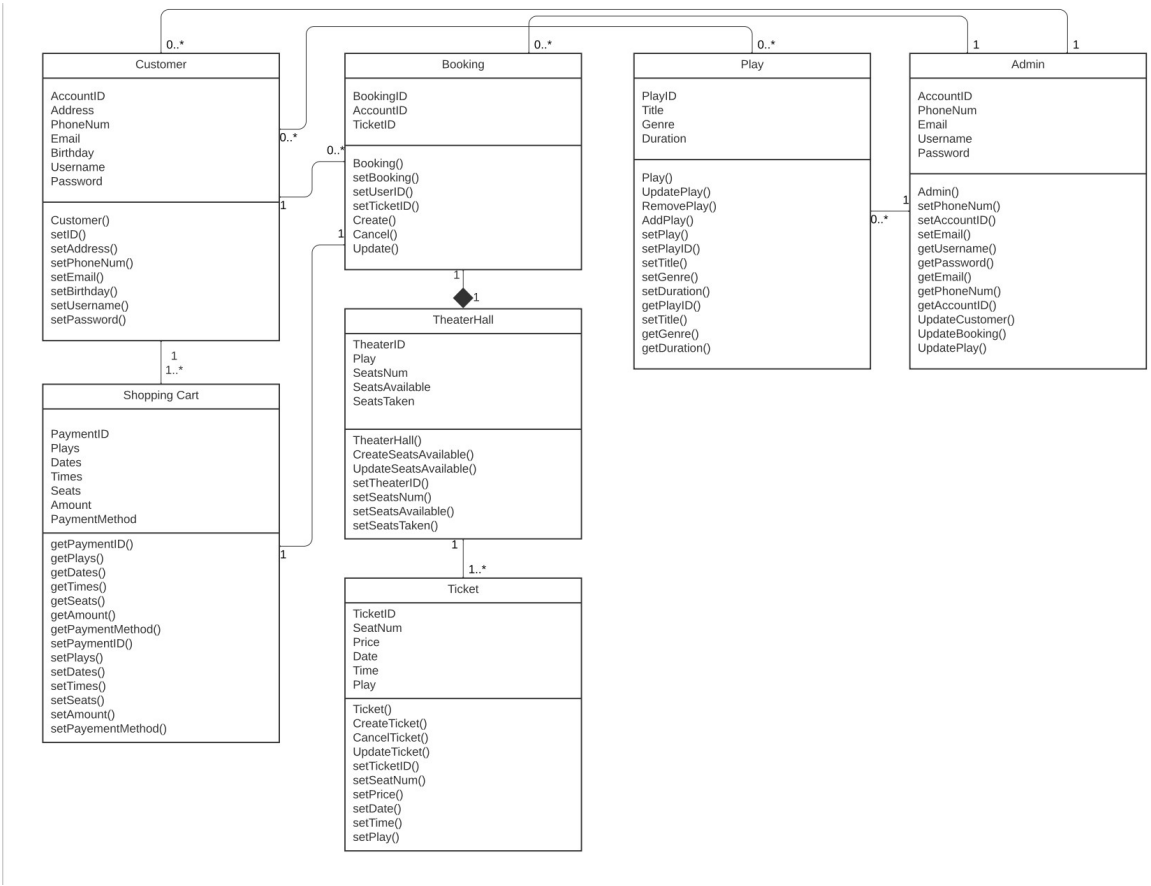


Figure 3 – Class Diagram

3.3 Detailed Non-Functional Requirements

3.3.1 Logical Structure of the Data

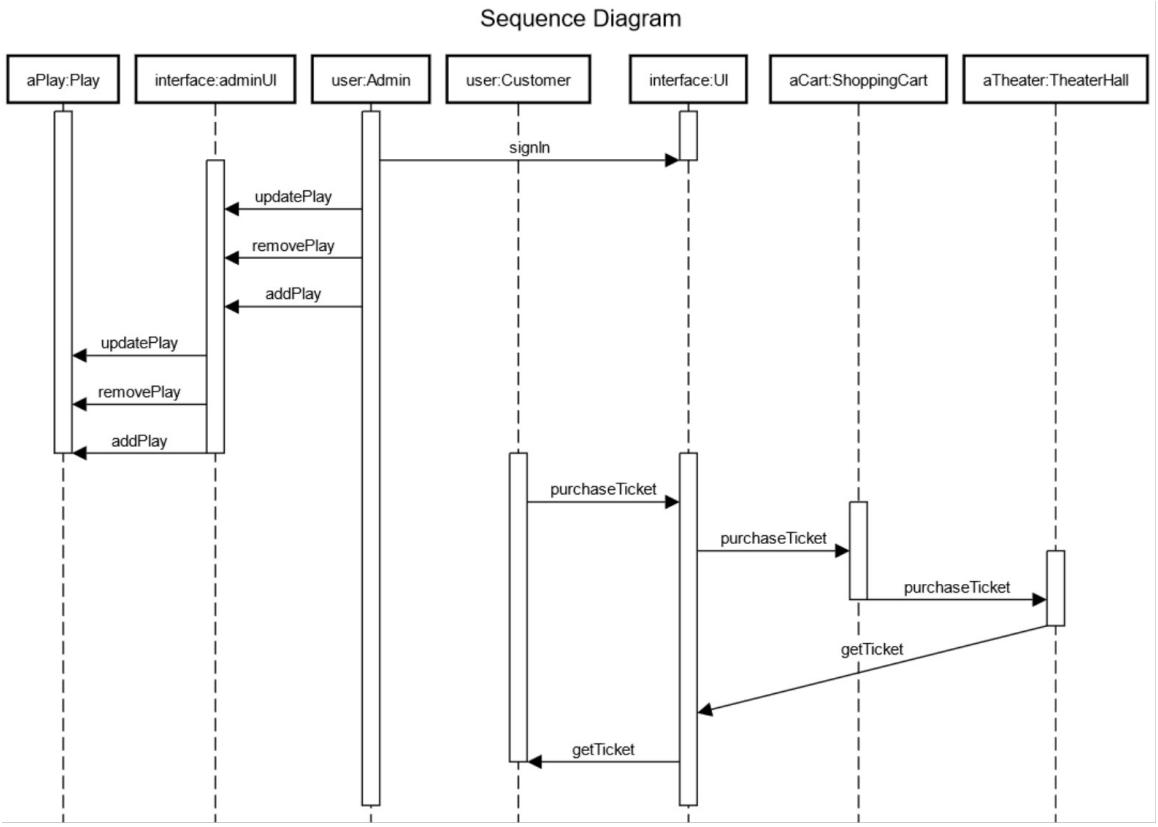


Figure 2 – Sequence Diagram

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.