MIDNIGHT VIEWER

Software Requirements Specification

Version 1.0

9/18/2025

Group #11

Johnny Thai
Aaron Justin Garcia Cayanan
Ahmad Khwajazada

Prepared for
CS 250-Introduction to Software Systems
Instructor: Gus Hanna, Ph.D.
Fall 2025

Revision History

Date	Version	Description	Author
9/18/2025	1.0	SRS 1.0	Group-11
10/05/2025	2.0	SDS 1.0	Group-11

Document Approval

Signature	Printed Name	Title	Date
Group 11	Group-11	Software Requirements Specification	9/18/2025
	Dr.Gus Hanna	Instructor, CS 250	
Group 11	Group-11	Software Design Specification	10/05/2025

Table of Contents

Software Requirements Specification

1. Introduction

1.1 Purpose

The purpose of this system is to provide customers with a way to purchase and redeem movie tickets both through a website, mobile application and in person sales. The customers will be able to select from standard room which has sixty seats or IMAX room which will have one hundred seats. Will give access to certain employees to verify tickets for the customers.

1.2 Scope

The system will let customers browse movies, seats and purchase tickets.
The system will have a selection of seats for standard room.
The system will have a selection of seats for IMAX room.
The system will provide digital tickets through QR code and apple wallet.
The system will have seat reservations, and subscription plan.
The system will allow administrators to manage issues in real time.
The system will be scalable as more theaters are added.

1.3 Definitions, Acronyms, and Abbreviations

QR code – quick response code to validate tickets
Subscription plan – monthly payment of $10 a month or yearly with $110.
Standard room – 60 seats are available to choose from.
IMAX room – 100 seats are available to choose from.
POS – point of sale.

1.4 References

The references are:

> Midnight viewer’s use cases for the customers with the website/application.
> Midnight viewer’s use cases for the employee with the system.
> Midnight viewer’s use cases for the administrator with the system.

1.5 Overview

The remaining sections of this document will provide the Overall description, specific requirements including external interface requirement, functionality, use cases, classes/objects, and non-functional requirements.

2. General Description

The movie theater ticketing system will serve customers, employees, and administrators. Customers will purchase tickets through the app, website, or at the theater.
Employees will sell tickets onsite and validate tickets at the theater entrance.
Administrators will manage refunds, troubleshoot technical issues, and oversee theater performance. The system will operate on both mobile and desktop platforms, support English and Spanish, and integrate with Apple Wallet and Google pay. All tickets and user data will be stored in a centralized database that will support multiple theater locations.

2.1 Product Perspective

The system will allow the customers to select seats and buy tickets.

2.2 Product functions

The system will have a scannable QR code, printable tickets, and have the users selecting seats and different tickets.

2.3 User Characteristics

The system will have users that are paying customers who like to watch films.
The system will support various genres for user enjoyment.
The system will support latest and most trending movies.
The system will support purchasing of collector items like popcorn buckets and movie drink cups.

2.4 General Constraints

The system will require license for movies.
The system will require the developer to adhere to the documentation for necessary external systems such as payment systems / payment providers.

2.5 Assumptions and Dependencies

The system will be able to support an offline mode for in movie theater purchasing systems that will have access to the database, in the event the internet goes down. This offline mode will be a standalone desktop application.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 Users Interfaces

The system will have a web interface.
The system will have a mobile application.
The system will have an Employee POS terminal.
The system will have an Administrator dashboard.

3.1.2 Hardware Interfaces

The system will have QR scanners at entrances.
The system will have point of sale hardware for employees.
The system will have Server hardware with redundancy.

3.1.3 Software Interfaces

The system will have payment APIs with Google pay and Apple pay
The system will have an Authentication method with Google open authorization.
The system will have ticket storage in a database linked to mobile/web application.

3.1.4 Communication interfaces

The system will have all communication secured with HTTPS/TLS
The system will have mobile notification for ticket confirmation
The system will have Email/SMS confirmation for purchases.

3.2 Functionality

3.2.1 Graphical User Interface

The system will have an interface for both desktop and mobile.
The system will be built on Expo for cross-platform compatibility.
The system will allow users to browse movies, select seats, and pay accordingly.

3.2.2 Accessibility

The system will support English and Spanish.
The system will be screen reader compatible.

3.2.3 Ticket Purchasinng

The system will allow ticket purchases through app and websites.
The system will allow one user to purchase 10 tickets per transaction.
The system will have regular and deluxe tickets depending on location.
The system will allow tickets to be refundable up to 48 hours before showtime.
The system will allow tickets to be purchased until 30 minutes after showtime begins.

3.2.4 Ticket Formats

The system will allow tickets to be in apple wallet, QR code, or printable in black and white.

3.2.5 User Accounts

The system will allow customers to login to the website, and app with the same username and password.
The system will have a subscription option of $10 per month or $110 per year with 10% off tickets and priority snack line for students, seniors, and veterans.

3.2.6 Discounts

The system will be able to verify age, student, and veteran status through ID verification.
The system will give students, seniors, and veteran discounts of 10%.

3.2.7 Seat reservations

The system will allow customers to select seats.
The system will have a section for standard room which will have 60 seats.
The system will have a section for IMAX room which will have 100 seats.

3.2.8 Scalability & Concurrency

The system will support up to 15,000 concurrent users during peak hours.
The system will be a centralized database that supports multiple theaters.
The system will have real-time updates for ticket availability and reservations

3.2.9 Payments & Security

The system will support credit cards, debit cards, Apple pay, and Google Pay.
The system will have all payments using SSL/TLS encryption
The system will comply with PCI DSS standards.
The system will have QR code validated at check-in to prevent fraud.

3.2.10 System Features

The system will have reserved tickets held for 10 minutes then made available if the user did not complete the purchase.
The system will allow users to refund their tickets 48 hours before showtime.
The system will have admin mode for the employees so that they can troubleshoot purchases and seat issues.
The system will have a feedback option with surveys after showtimes.

3.3 Use Cases

3.3.1 Use Case#1: Customer buys the tickets

The system will allow the customer to select seats and buy the tickets

3.3.2 Use Case#2: Employee processing the tickets

The system will allow the employee to process the tickets for the customers.

3.3.3 Use Case#3: Administration resolving issues

The system will all for admin to process refunds and reprint tickets.

3.4 Classes/Objects

3.4.1 customers/payment

The system will have ID, username, password, email payment information, discount eligibility, subscription status.

3.4.2 Employee/Email/SMS

The system will have employee ID, role, POS access for the employees
The system will be able to process sales, issue printed tickets and send SMS confirmation.

3.4.3 Administration/System

The system will have admin ID, and privileges for administration
The system will be able to manage refunds, subscriptions, and system monitoring.

3.5 Non-functional requirements

3.5.1 Performance

The system will load the page under three seconds for normal use.
The system will have payments and tickets processed within five seconds.
The system will have real-time updates for seat reservations and availability.

3.5.2 Reliability

3.5.2.1 Back-end Internal Computers

The system will have a server that will maintain 99.9% uptime.
The system will have back-up servers for high availability.

3.5.3 Availability

3.5.3.1 Internet Service Provider

The system will have a failover ISP connection for reliability.

3.5.4 Security

3.5.4.1 Data Transfer

The system will have all data transfer encrypted with TLS.
The system will not have sensitive data transmitted in plaintext.

3.5.4.2 Data Storage

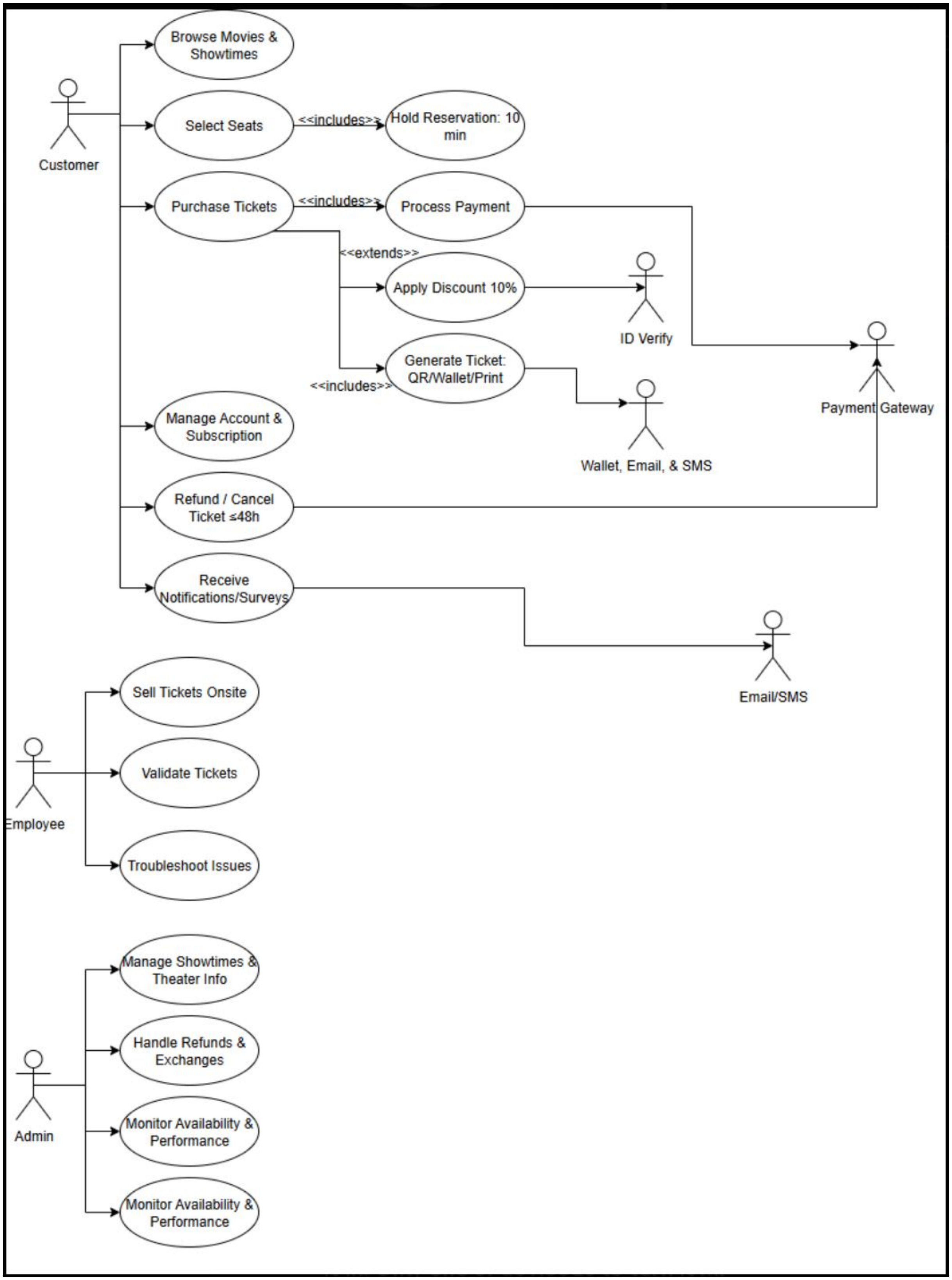
The system will have passwords stored with bcrypt hashing and salting.
The system will have PII stored according to GDPR and U.S. privacy standards.
The system will have ticket validation data stored in encrypted databases.

3.5.5 Maintainability

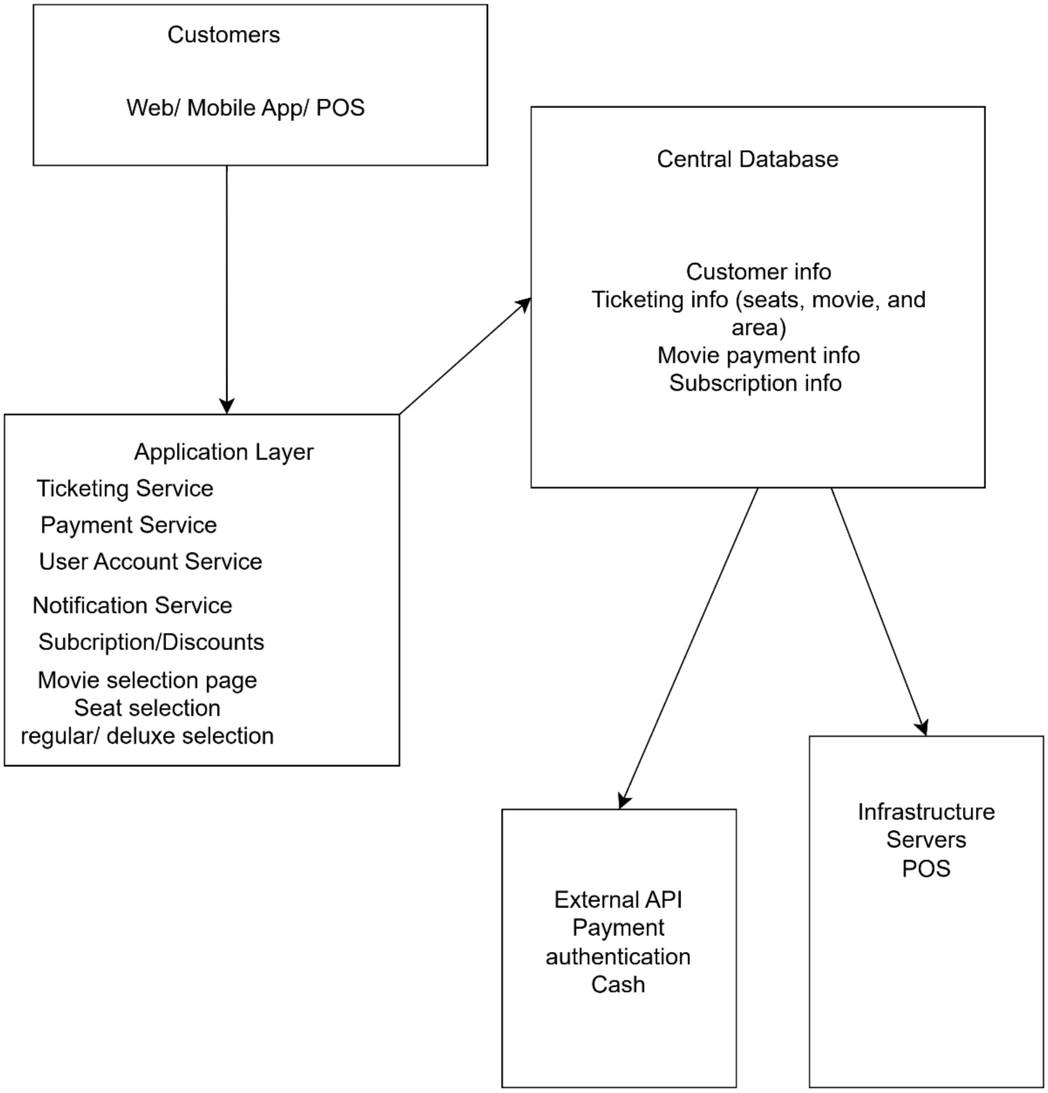
The system will have an admin portal.
The system will be created using modular components for easy fixing/debugging

3.5.6 portability

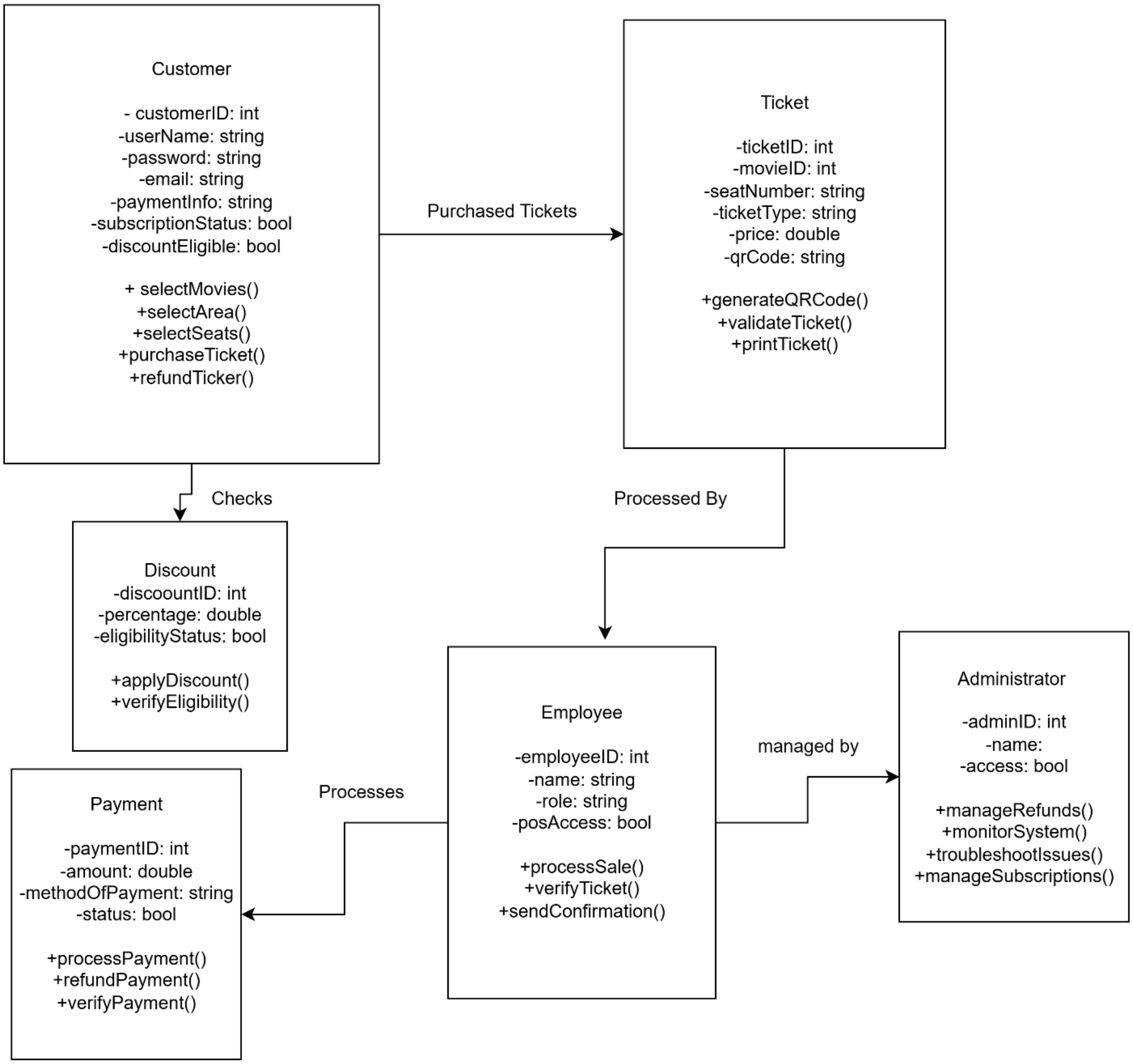
The system will be a web app and will be access through internet.
The system will be a stand-alone executable application in case the internet goes down.
(this will act as a fall-back system)


Customer
Browse Movies & Showtimes
Select Seats
<<includes>>
Hold Reservation: 10 min
Purchase Tickets
<<includes>>
Process Payment
<<extends>>
Apply Discount 10%
ID Verify
Generate Ticket: QR/Wallet/Print
<<includes>>
Wallet, Email, & SMS
Payment Gateway
Manage Account & Subscription
Refund / Cancel Ticket ≤48h
Receive Notifications/Surveys
Email/SMS
Employee
Sell Tickets Onsite
Validate Tickets
Troubleshoot Issues
Admin
Manage Showtimes & Theater Info
Handle Refunds & Exchanges
Monitor Availability & Performance
Monitor Availability & Performance

- Architectural diagram of all major components



- UML Class Diagram



4.3.1 Description of Customer Class

Customer- Lowest level of access to the system. Can browse showtimes, select seats, checkout, manage tickets/refunds, subscriptions, and account preferences. This class will hold the following attributes

-customerID: int - Unique ID for the customer.

-userName: string - Login/display name of the customer.

-password: string - Hashed password for authentication.

-email: string → Contact and verification method.

-paymentInfo: string → Stored payment method (e.g., encrypted card info).

-subscriptionStatus: bool → Indicates if the customer is subscribed to a plan.

-discountEligible: bool → Marks whether customers qualify for discounts.

Methods:

+selectMovies(): will allow customers to select different movies types sorted by genre.

+selectArea(): will allow customers to select between regular or deluxe.

+selectSeats(): will allow customers to select different seats in regular or deluxe depending on what is chosen.

+purchaseTicket(): will allow customers to select the amount of tickets they want.

+refundTicket(): will allow customers to apply for refunds for the movie ticket purchased

4.3.2 Description of Ticket Class

Ticket- Represents a right of entry for a specific seat at a scheduled showtime. This class will hold the following attributes

-ticketID: int -Unique ID for the ticket.

-movieID: int - References the movie/showtime.

-seatNumber: string - Seat assignment in auditorium.

-ticketType: string - Standard, IMAX, Student, Senior.

-price: double - Final ticket price after discounts, fees.

-qrCode: string - Unique scannable code for validation

Methods:

+generateQRCode() : will create the QR code for the ticket that has been purchased

+validateTicket(): will link tickets to the correct customers

+printTicket(): will have a window where customers can print their tickets.

4.3.3 Description of Discount Class

Discount- Applies promotional reductions, student/senior eligibility, ensures eligibility rules are enforced. This class will hold the following attributes

-discountID: int – unique ID for different customers

-discountPercentage: double – value for the discount

-eligibilityStatus: bool – true if customer is eligible for the discount, false if otherwise.

Methods:

+applyDiscount() – if customer is eligible for the discount apply the discount if not do not apply discount.

+verifyEligibility()- checks if the customer is eligible or not

4.3.4 Description of Payment Class

Payment- processes the payment method of the customers. This class will hold the following attributes

-paymentID: int – provides the customer with a unique paymentID for access in the future if customer needs refund

-amount: double – the cost of the tickets

-methodOfPayment: string – name of the payment method ex. Amex. Bank of America or chase

-status: bool – customer purchased or not.

Methods:

+processPayment() – will process the payment for the movie and seat selected for the customer.

+refundPayment()- will allow customers to select refund during a certain period. Refundable up to 48 hours before showtime.

+verifyPayment()- allows customers to confirm their payments of the tickets and verify if it is available.

4.3.5 Description of Employee Class

Employees- keeps track of the employees and their status within the company. Represents box office or front-line staff. This class will hold the following attributes

employeeID: int - Unique staff ID.

name: string - Employee name.

role: string - Role description (e.g., “Cashier,” “Usher”).

posAccess: bool - Determines if they can access POS (Point of Sale) system.

Methods:

+processSales() – will allow employees to select tickets for the customers in person.

+verifyTickets()- will allow employees to match tickets with names of customers.

+sendConfirmation() – will allow employees to send email confirmation of the tickets after purchase is made in person.

4.3.6 Description of Administrator Class

Administrator - Back-office operators with the highest level of access. Manages the overall operation of the system itself. Full access is granted. This class will hold the following attributes

adminID: int - Unique admin ID.

name: string - Administrator name.

access: bool - Flag indicating active access permissions.

Methods:

+manageRefunds()- will allow admin to process refunds for the customers

+monitorSystem()- will allow admin to check the inflow of the system and shut down if needed.

+troubleshootIssues()- will allow admin to manually add tickets if something goes wrong with the ticket purchasing.

+manageSubscriptions()- will allow admin to get rid of subscriptions or add subscriptions to the customers.

Overview of the Timeline and Tasks

Overview of the software system- the software system is comprised of different components such as a website, mobile app, and POS terminal. Each component will interact with the centralized server and have a shared database. It will manage users, tickets, payments, and schedule information. The frontend component will allow customers to select different areas regular or deluxe, movies, seats. There will be a secure payment method for the movies selected using Apple Pay and Google Pay. The POS system will have real-time access to the ticket validations. The system will use encrypted HTTPS/TLS connections, ensuring security.

List of tasks

2 weeks duration - Web interface- design the web application for the ticketing system – Johnny Thai

2 weeks duration - Mobile application – create the mobile application using Expo for IOS and android – Aaron Garica

3 weeks duration - Backend Services – implement APIs for authentication, subscriptions, and data processing – Ahmad Khwajazada

2 weeks duration - Database setup- manage database for users, tickets, and payments – Aaron Garica

1 weeks duration - Payment Integration – integrate Google Pay, Apple Pay, and card processing securely – Johnny Thai

1 weeks duration - Employee POS system – build interface for ticket validation, refunds, and sales at physical locations – Ahmad Khwajazada

1 weeks duration - Administrator Dashboard – Develop admin tools for refunds, performance tracking, and system monitoring.

1 weeks duration - Testing and debugging – conduct testing of the full system- All members

1 weeks duration Documentation and reporting – maintain SRS documentation, UML diagrams, and user manuals – Johnny Thai