Movie Theater Ticketing System

Software Design Specification
5/01/24

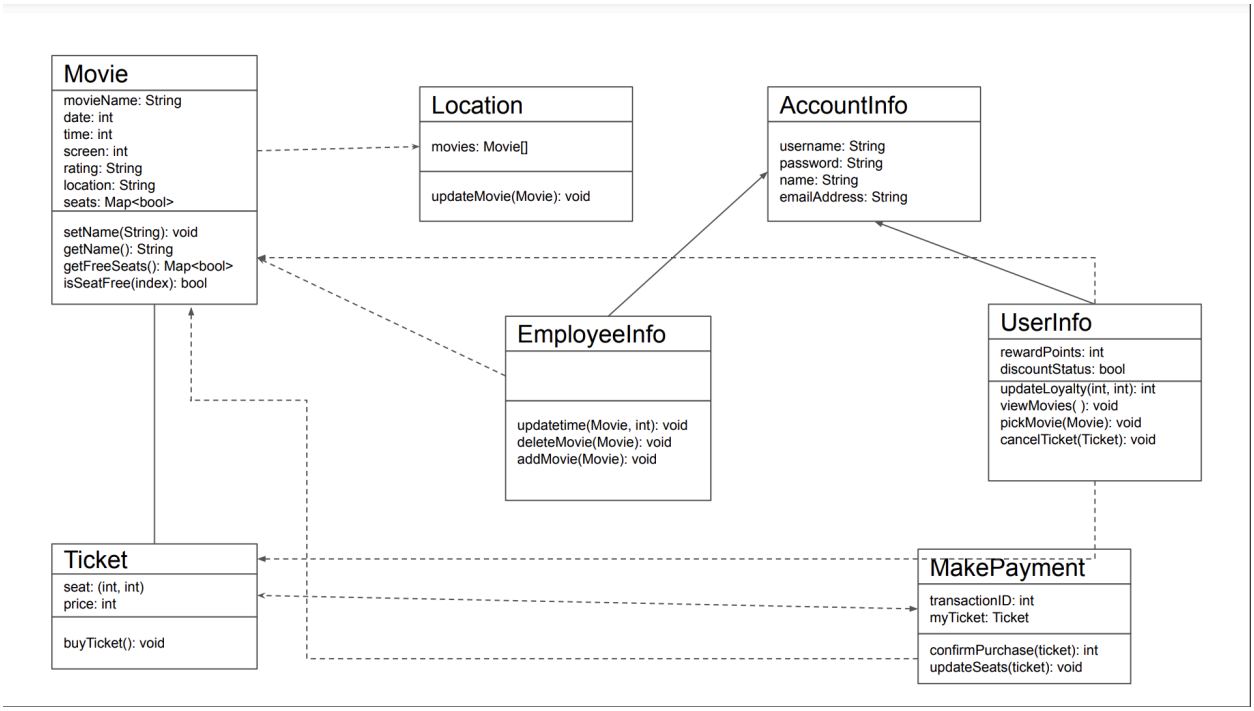
Group #4
Liam Hayes, Youngmin Park, Alex Colmenar

Prepared for
CS 250- Introduction to Software Systems
Instructor: Gus Hanna, Ph.D.
Spring 2024

Brief Overview:

The Movie Theater Ticketing System focuses on the stakeholders and applications that allow for online ticket sales, distribution, and marketing of movies. It will provide an accessible and intuitive interface for users to purchase tickets through a browser. The system will ensure it manages showtimes, ticket availability, and user transactions, ensuring consistency and security for the different movies. It will display movie reviews from critics updated constantly for live reviews and will not produce its reviews.

UML Diagram:



Description:

Class: Movie

- Attributes:
  - `movieName: String` - The name of the movie.
  - `date: int` - The showing date of the movie.
  - `time: int` - The showing time of the movie.
  - `screen: int` - The screen number where the movie is shown.
  - `rating: String` - The rating of the movie (PG, PG-13, R).

    - `location: String` - The location of the movie showing.
    - `seats: Map<bool>` - A map representing the seats and their availability (true for available, false for taken).
- Operations:
    - `setName(String): void` - Sets the name of the movie.
    - `getName(): String` - Retrieves the name of the movie.
    - `getFreeSeats(): Map<bool>` - Returns a map of seats with their availability status.
    - `isSeatFree(index): bool` - Checks if a specific seat is free.

Class: Ticket

- Attributes:
    - `seat: (int, int)` - The row and column of the seat.
    - `price: int` - The price of the ticket.
- Operations:
    - `buyTicket(): void` - Processes the ticket purchase.

Class: Location

- Attributes:
    - `movies: Movie[]` - An array of `Movie` objects available at the location.
- Operations:
    - `updateMovie(Movie): void` - Updates the movie details.

Class: AccountInfo

- Attributes:
    - `username: String`
    - `password: String`
    - `name: String`
    - `emailAddress: String`

Class: EmployeeInfo

- Operations:
    - `updateTime(Movie, int): void` - Updates the time for a movie showing.
    - `deleteMovie(Movie): void` - Removes a movie from the schedule.
    - `addMovie(Movie): void` - Adds a new movie to the schedule.

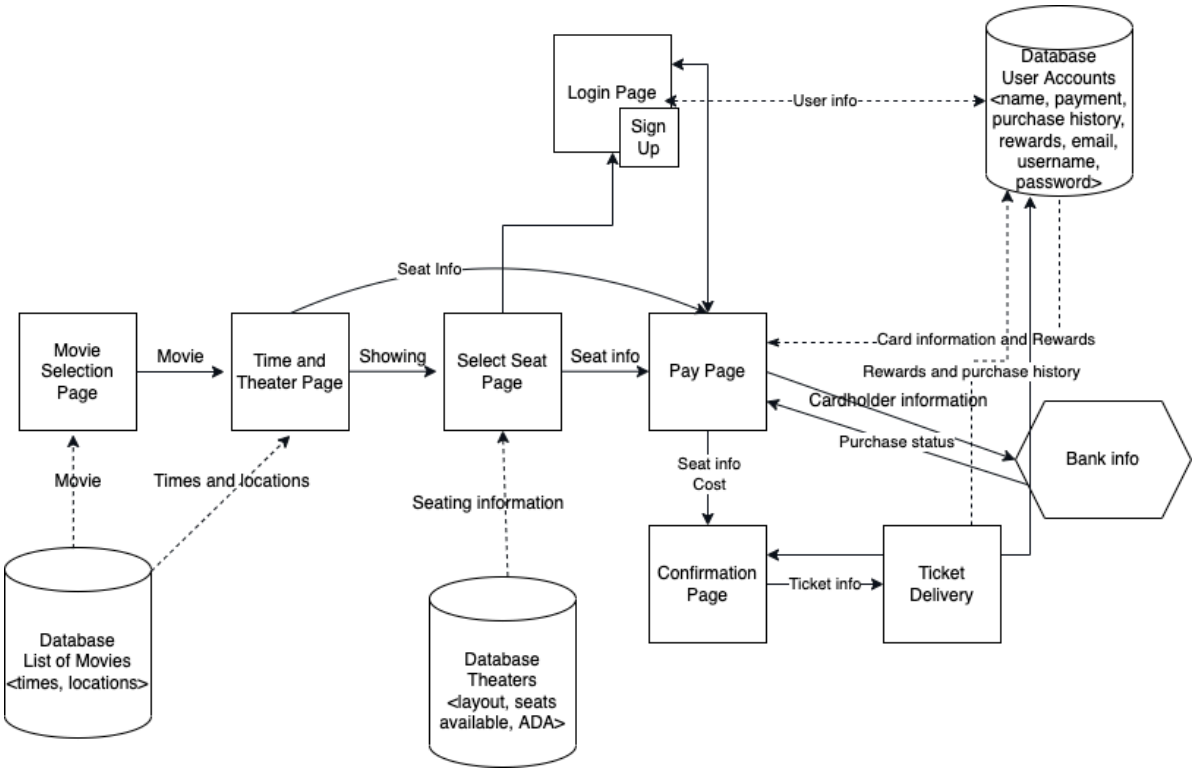
Class: UserInfo

- Attributes:
  - `rewardPoints: int` - The loyalty points of the user.
  - `discountStatus: bool` - Indicates if the user is eligible for a discount.
- Operations:
  - `updateLoyalty(int, int): int` - Updates loyalty points for the user.
  - `viewMovies(): void` - Displays available movies.
  - `pickMovie(Movie): void` - Selects a movie for ticket purchase.
  - `cancelTicket(Ticket): void` - Cancels a ticket purchase.

Class: MakePayment

- Attributes:
  - `transactionID: int` - The ID of the transaction.
  - `myTicket: Ticket` - The ticket associated with the payment.
- Operations:
  - `confirmPurchase(ticket): int` - Confirms the purchase of a ticket and returns a transaction ID.
  - `updateSeats(ticket): void` - Updates the seat availability after a ticket purchase.

SWA Diagram:



Description:

Movie Selection Page: Users start here to browse movies. This component interacts with a database to retrieve a list of movies, including their showtimes and locations.

Time and Theater Page: After selecting a movie, users are directed here to choose a specific time and theater location, pulling data from a database that lists movie times and locations.

Select Seat Page: Users pick their seats based on availability, which is checked against a seating database for the selected theater.

Pay Page: Payment information is collected and processed in this component, with secure transactions being a priority.

Confirmation Page: This component confirms the successful transaction, providing users with their ticket information and seat details.

Ticket Delivery: Manages the distribution of tickets, ensuring users receive them in their chosen format.

The databases—**User Accounts**, **List of Movies**, and **Theaters**—store all relevant data and interact with the system's pages to provide up-to-date information and maintain the integrity of user data and transactional records.

The **Login Page** and **Sign Up** are connected to the **User Accounts** Database, handling user authentication and account creation. Additionally, the system interfaces with external **Bank Info** for payment processing, ensuring secure financial transactions.

Connectors between these components represent data flow and user navigation paths, indicating how users move through the system and how data is exchanged to support functionalities like movie selection, seat reservation, and payment processing.

Development Plan and Timeline

- **Partitioning of Tasks:**
    - UI Design and Development
    - Backend System Development
    - Payment Integration
    - Testing and Quality Assurance
- **Team Member Responsibilities:**

    - Alex Colmenar: UI Design and Frontend Development
    - Liam Hayes: Backend System Development, Testing and Quality Assurance
    - Youngmin Park: Database Management and Integration
- **Timeline:**
    - Week 1-2: Requirement Analysis and Planning
    - Week 3-4: UI Design
    - Week 5-8: Backend Development
    - Week 9-10: Integration and Testing
    - Week 11: Final Review and Deployment

Verification Test Plan

Introduction to Testing

- The purpose of this test is to verify all components of the Ticketing System function as expected. We want to ensure a seamless user experience, tight security, and data integrity.

Test Strategy

- The test plan will use a multi-tiered testing plan, including functional tests, software system tests, and unit tests to cover the entirety of the system.
- It would implement automated and manual testing to ensure efficiency and coverage over every problem that could occur within the system.

Test Levels

- Unit Testing: Focus on testing the smallest pieces of code and make sure each function has the correct and intended output.
- Functional Testing: Focus on testing specific functions within the software system.
- System Testing: Test the system as a whole and make sure it runs as expected on the surface and is ready to be used.

Test Cases

- Each case follows a structured approach to make sure it has thoroughly found any errors and provides clarity in how the test is done. In total this covers the unit, functional, and system levels of the Movie Theater Ticketing System.

Failure Management

- Provides a plan for an analysis of the failed areas of the system to find and understand the root cause of the issues at hand. This not only solves the critical issues, but being documented will help refind the SDLC preventing any similar issues from occurring in future development.

Book.xlsx

Data Management Strategy

Our data management strategy employs a hybrid database approach to provide the necessary balance between structured data management, scalability, and flexibility. We incorporate both SQL and NoSQL databases to leverage their strengths and ensure optimal performance.

SQL Databases:

For transactional data that requires relational integrity and complex querying, such as user account details, ticket transactions, and showtime schedules, we use SQL databases. They offer ACID properties, ensuring reliable and consistent transactions necessary for the integrity of our system.

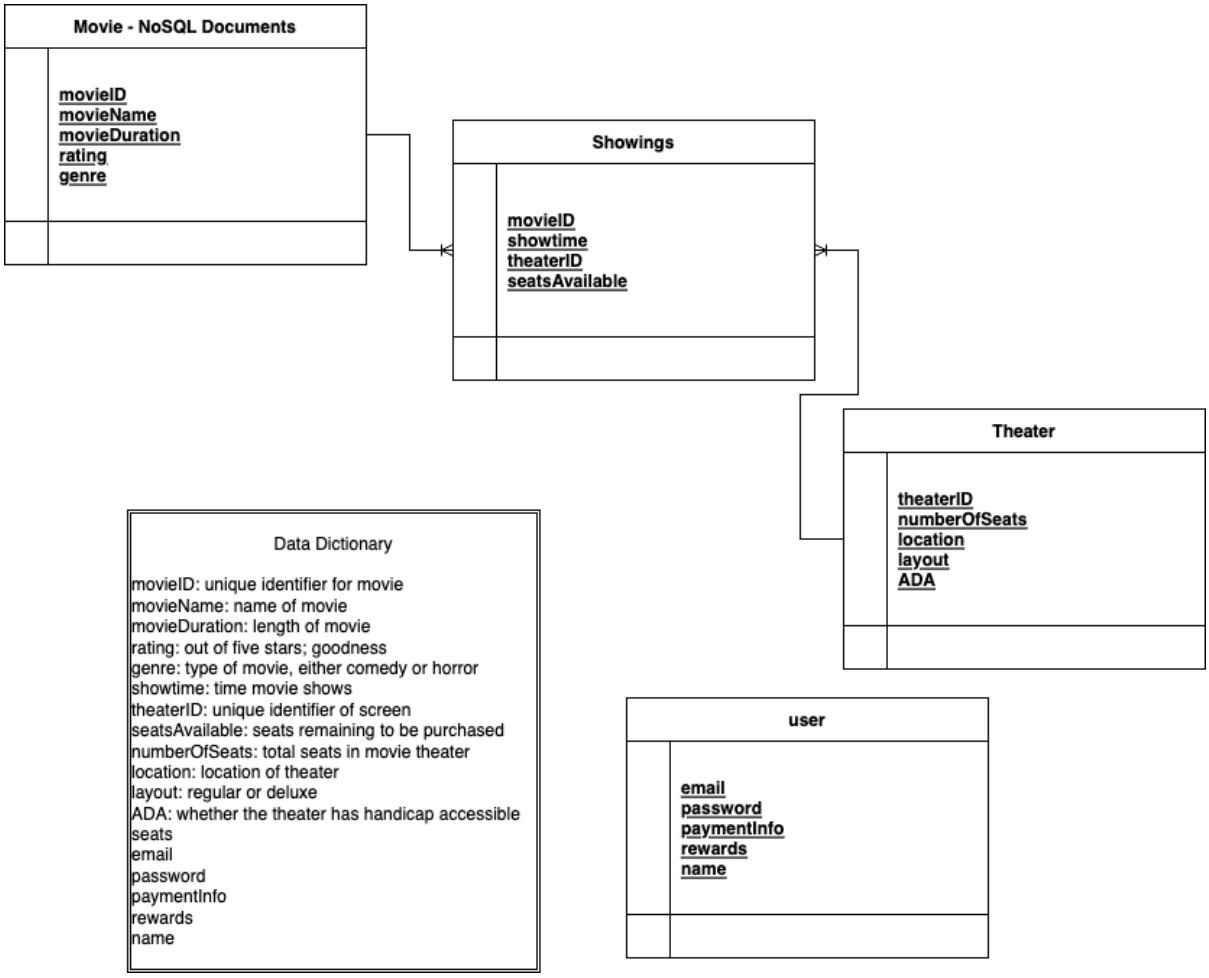
One key component of our SQL database is the Theater Information Table. This allows us to efficiently manage theater-specific information, which is crucial for seat selection and maintaining an accurate representation of each theater's capabilities and layout.

VID	Theater ID	# of seats	Layouts	ADA (bool)	Location
LMD1	01	20	Regular	y	La Mesa
LMD2	02	10	Deluxe	y	La Mesa
	01	50	Regular	y	El Cajon
	02	100	Regular	y	El Cajon

Furthermore, our system features a Movie Database. This design facilitates the relationship between movies and their showtimes at various theaters, while also providing users with valuable information for making informed decisions on movie selections.

Name	Showtime	Rating	Genre	Duration	Theater ID	Available Seats
Minions	7 pm	5 Stars	Comedy	91 min.	01	1
Batman	7 pm	4.5 Stars	Thriller	176 min.	02	10
Minions	9 pm	5 Stars	Comedy	91 min.	01	0

Entity-Relationship Diagram for SQL Documents:



This Entity-Relationship Diagram represents the structure of the SQL documents within our system. It highlights the data dictionary for the movie entities, showings, and theater information. The attributes within these documents include unique identifiers, movie details, showtime specifics, and theater attributes. This design supports rapid retrieval and updating of movie and theater data, which is essential for our application's performance and user experience.

NonSQL Databases:

To manage semi-structured or unstructured data like movie reviews, user comments, and potential high-velocity data, we integrate NoSQL databases. These databases provide scalability and the flexibility to store data without a predefined schema, accommodating a variety of data types and structures.

Security Measures:

We enforce strict security protocols, including data encryption at rest and in transit, to protect sensitive user information and payment details. Our APIs, secured with OAuth, ensure that data access is tightly controlled and only authorized operations are permitted.

Backup and Recovery:

A robust backup and recovery strategy is in place to protect against data loss and to ensure high availability. Regular, encrypted backups are taken, and our infrastructure is designed to allow for quick recovery in the event of a failure.

Trade-offs and Alternatives:

While the hybrid database approach provides both flexibility and performance, it also introduces complexity in managing and integrating multiple systems. We have considered this in our design decisions and believe that the benefits, such as improved scalability and flexibility, outweigh the costs. Alternative approaches, such as using a single versatile database system, were considered but found to be less optimal for our use case.

In conclusion, our data management strategy is designed to be dynamic, secure, and efficient, supporting the needs of our Movie Theater Ticketing System and ensuring a seamless and responsive user experience.

Verification Test Plan

Introduction to Testing

The purpose of this test is to verify all components of the Ticketing System function as expected. We want to ensure a seamless user experience, tight security, and data integrity.

Test Strategy

- The test plan will use a multi-tiered testing plan, including functional tests, software system tests, and unit tests to cover the entirety of the system.
- It would implement automated and manual testing to ensure efficiency and coverage over every problem that could occur within the system.

Test Levels

- Unit Testing: Focus on testing the smallest pieces of code and make sure each function has the correct and intended output.

- Functional Testing: Focus on testing specific functions within the software system.
- System Testing: Test the system as a whole and make sure it runs as expected on the surface and is ready to be used.

Test Cases

- Each case follows a structured approach to make sure it has thoroughly found any errors and provides clarity in how the test is done. In total this covers the unit, functional, and system levels of the Movie Theater Ticketing System.

Failure Management

- Provides a plan for an analysis of the failed areas of the system to find and understand the root cause of the issues at hand. This not only solves the critical issues, but being documented will help refind the SDLC preventing any similar issues from occurring in the future development

Movie Theater Ticketing System

Software Requirements Specification

Group #4
Liam Hayes, Youngmin Park, Alex Colmenar

Prepared for
CS 250- Introduction to Software Systems
Instructor: Gus Hanna, Ph.D.
Spring 2024

Revision History

Date	Description	Author	Comments
2/11/24	Version 1	LH, YP, AC	First Revision
2/13/24	Version 2	LH, YP, AC	Second Revision
2/14/24	Version 3	LH, YP, AC	Third Revision
2/16/24	Version 4	LH, YP, AC	Fourth Revision (Restart)

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	Liam Hayes	Software Eng.	
	Dr. Gus Hanna	Instructor, CS 250	

Table of Contents

1. Introduction

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS. It includes the overall purpose, definitions, references, and broad overview of the SRS. The SRS aims to provide an in-depth insight into the complete **Movie Theater Ticketing System** by describing and analyzing the problem statement in depth. Furthermore, it provides the requirements set by the stakeholders and their needs for this product's features. The detailed requirements of **Movie Theatre Ticketing System Software** are provided within this document.

1.1 Purpose

The purpose of the document is to collect data concerning consumers and analyze each idea that defines the system and the requirements. Also, we will include how we hope this SRS will be used to gain a better understanding of the outlined concepts that may be edited or developed in the future.

In short, the purpose of this SRS document is to provide a detailed overview of our software product, its goals, and parameters. This document describes the target audience and its requirements along the lines of its user interface, hardware, and software systems. The software system will describe how our team and consumers utilize the product's functionality. Our document as an entirety is to help any designer or developer to assist in the tech software delivery lifecycle (SDLC).

1.2 Scope

The scope pertains to an online Movie Theater Ticketing System. It focuses on the stakeholders and applications that allow for online ticket sales, distribution, and marketing of movies. It will provide an accessible and intuitive interface for users to purchase tickets through a browser.

This SRS also specifies the requirements of the software needed to develop the online system. The goal of the system is to be easy and convenient for the buyer. The system will ensure it manages showtimes, ticket availability, and user transactions, ensuring consistency and security for the different movies. It will display movie reviews from critics updated constantly for live reviews and will not produce its reviews.

1.3 Definitions, Acronyms, and Abbreviations

Customer Service	Constant tech support for unplanned problems among users
UI/UX	User Interface / User Experience
FAQ	Frequently asked questions

API	Application Programming Interface

1.4 References

1. Theater Ticketing Requirements page on canvas
2. https://www.academia.edu/35876986/SOFTWARE_REQUIREMENTS_SPECIFICATION_FOR_MOVIE_BOOKING_SYSTEM

1.5 Overview

The remaining sections of this document will provide a general description of the product including, focusing on the buyer, what hardware would be needed, and lastly the data requirements/functionality of the product. The general description will be discussed in section 2. The requirements and functionality of the product will all be discussed in section 3. Also showing a viewpoint of the website. Section 4 is then for the supporting information of the website.

2. General Description

This document contains the problem statement that the current system is stunting the growth opportunities of the company. Furthermore, it brings up the proposed solution illustrated by the stakeholders during their brainstorming and the requirements to meet their needs. This document further describes the features of the software system and a description of each system proposed by the stakeholder.
The following SRS contains the requirements of the stakeholders and provides a detailed function for the online store, including its assumptions, constraints, dependencies, and requirements.

2.1 Product Perspective

The Theater Ticketing System is a standalone product designed to serve as the primary interface for ticket sales, performance scheduling, and customer management for theaters. This system is an integral component of the theater's overall operational infrastructure, interfacing with several external and internal systems to provide a seamless and efficient ticketing experience.

2.2 Product Functions

The Theater Ticketing System software will facilitate several key functions to enhance the experience for both the theater staff and patrons. Key functionalities include:

Ticket Sales: Allows users to purchase tickets online for current and future performances. It supports seat selection based on real-time availability.

Performance Scheduling: Enables theater staff to schedule performances, including date, time, and details about the show.

Customer Management: Manages customer information, including purchase history and preferences, to enhance customer service and marketing efforts.

Seat Management: Provides a detailed layout of the theater seating plan, allowing for dynamic pricing and seat availability updates.

Payment Processing: Integrates secure payment processing for ticket sales, refunds, and exchanges.

Reporting: Generates reports on sales, and customer demographics, and shows popularity to aid in strategic planning and marketing.

2.3 User Characteristics

The Theater Ticketing System is designed for two main user groups with distinct characteristics:

Theater Staff: This includes administrators, ticketing agents, and marketing personnel. They require a system that is efficient, reliable, and easy to use for managing performances, ticket sales, customer information, and reporting. Their interaction with the system is extensive, necessitating a user-friendly interface with robust functionality for scheduling, sales management, and data analysis.

Customers: End-users looking to purchase tickets. They range from tech-savvy individuals who prefer online transactions to those who may require more traditional methods of engagement. The system must be accessible, intuitive, and secure, providing a seamless experience from browsing to purchasing tickets, with options for seat selection and payment processing.

2.4 General Constraints

The Theater Ticketing System will operate under several constraints that affect the design and development of the software:

Technology Compatibility: The system must be compatible with the existing hardware and software infrastructure of the theater. This ensures smooth integration and operation without extensive modifications to current systems.

Budget Constraints: The development and operational costs of the system should not exceed the budget allocated by the theater. This includes costs related to software development, maintenance, and any third-party services or integrations required for payment processing and data management.

User Accessibility: The system must be accessible to a wide range of users, including those with disabilities. It should adhere to accessibility standards to ensure that all patrons can easily navigate and use the ticketing system.

Scalability: The system must be scalable to accommodate future growth in terms of the number of performances, ticket sales, and users. It should support increased loads without significant degradation in performance or user experience.

Operational Timeframe: The system's development and implementation must be completed within a predefined time frame to meet the theater's scheduling and operational requirements. Delays in deployment could affect performance scheduling and ticket sales.

2.5 Assumptions and Dependencies

Hardware Availability: It is assumed that the theater's current hardware infrastructure is sufficient to support the new ticketing system. This includes servers, workstations, and networking equipment.

Software Environment: The system assumes the availability of a specific software environment, including operating systems, database management systems, and web servers.

Internet Connectivity: Continuous internet connectivity is assumed for online ticket sales and access to cloud-based services. Any limitations in connectivity or bandwidth could impact the functionality and responsiveness of the ticketing system.

Third-Party Services: The system depends on third-party services for payment processing, email notifications, and possibly cloud storage.

Regulatory Environment: The system assumes compliance with current data protection and privacy laws.

User Behavior: Assume that the theater’s volume of concurrent users is manageable by the software at any time during the theater’s working hours.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.1.1 This system shall display a list of all of the movies to choose from

3.1.1.2 This system shall provide the user a choice of whether they want to interact with the movie and and read a small description of it

3.1.1.4 This system shall display the price of the ticket to watch the movie

3.1.1.5 This system shall enable the user to view past movies that were interacted with.

3.1.1.6 This system shall notify the user of the ratings of the movie if possible

3.1.1.7 This system shall display a button that could be interacted with to buy the movie

3.1.1.8 This system shall enable the user to put in their credit card information to buy the movie

3.1.1.9 This system shall provide the user with a receipt after purchasing a movie

3.1.2 Hardware Interfaces

3.1.2.1 The system shall use GPS to find possible nearby movie theaters that the buyer could attend to

3.1.3 Software Interfaces

3.1.3.1 This system shall have Payment processing integration with credit card services, PayPal, and Bitcoin.

3.1.3.2 This system shall have an email system for ticket delivery.

3.1.4 Communications Interfaces

3.1.4.1 This system shall integrate with email services to deliver tickets to customers.

3.1.4.2 This system shall have interfaces with digital kiosks for in-person ticket purchases and pickups.

3.2 Functional Requirements

3.2.1 Online Ticket Reservation

3.2.1.1 Introduction

This feature allows the user to buy movie tickets

The system shall have an interface with a database of showtimes and tickets available

The system allows administrators to alter the database of showtimes and tickets available

3.2.1.2 Inputs

The system shall ensure that the website can hold 1000 people at once

The system shall ensure it is a web browser

The system shall ensure tickets are only available for purchase 2 weeks prior to showtime and 10 minutes after showtime starts

3.2.1.3 Processing

The system shall support different discount tickets

The system shall only let a user buy 20 tickets max at a time

3.2.1.4 Outputs

The system shall provide a receipt after the purchase of a ticket

The system shall be able to scrape online review sites to display review and critic quotes of movie

3.2.1.5 Error Handling

The system shall block boat from trying to buy a lot of tickets in high demand movies

The system shall support administrator mode

The system shall have a customer feedback system

3.2.2 Discount and Promotion Management

3.2.2.1 Introduction

This system shall enable administrators to define and manage a variety of discount codes and promotional offers for ticket sales.

This system shall provide customers with the ability to apply discount codes and automatically receive promotional offers at checkout.

3.2.2.2 Inputs

This system shall accept discount codes entered by customers during the ticket-purchasing process.

This system shall allow administrators to input parameters for new promotions, including discount rates, eligibility criteria, and validity periods.

3.2.2.3 Processing

This system shall verify the validity and eligibility of discount codes against current promotions and user status.

This system shall calculate the final ticket price after applying the applicable discounts and promotions.

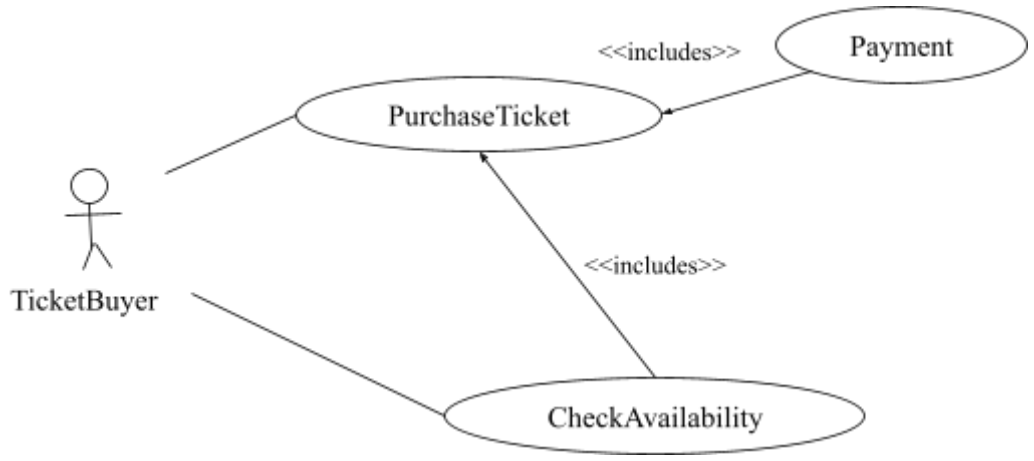
3.2.2.4 Outputs

This system shall display the final ticket price to customers, clearly showing the discounts applied.

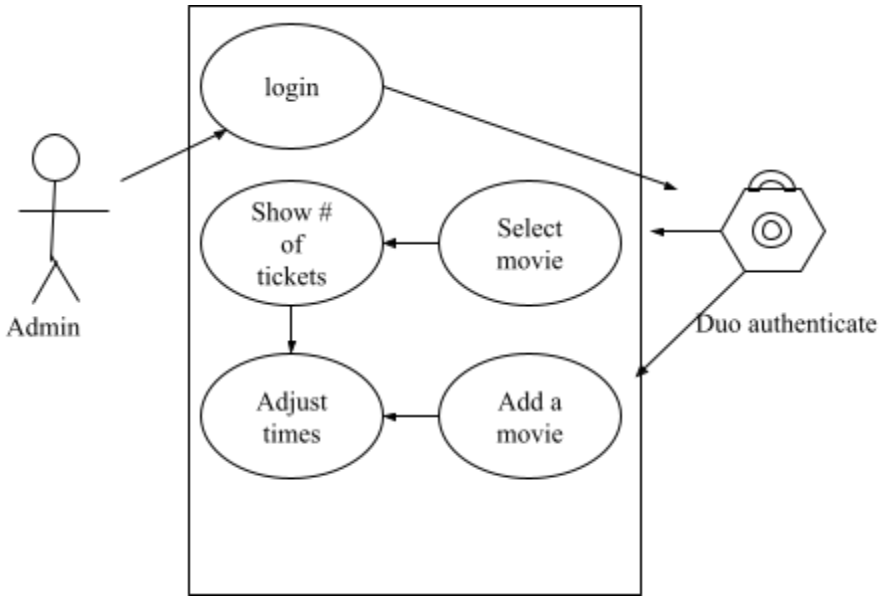
This system shall generate reports for administrators on the usage and impact of discounts and promotions on ticket sales.

3.3 Use Cases

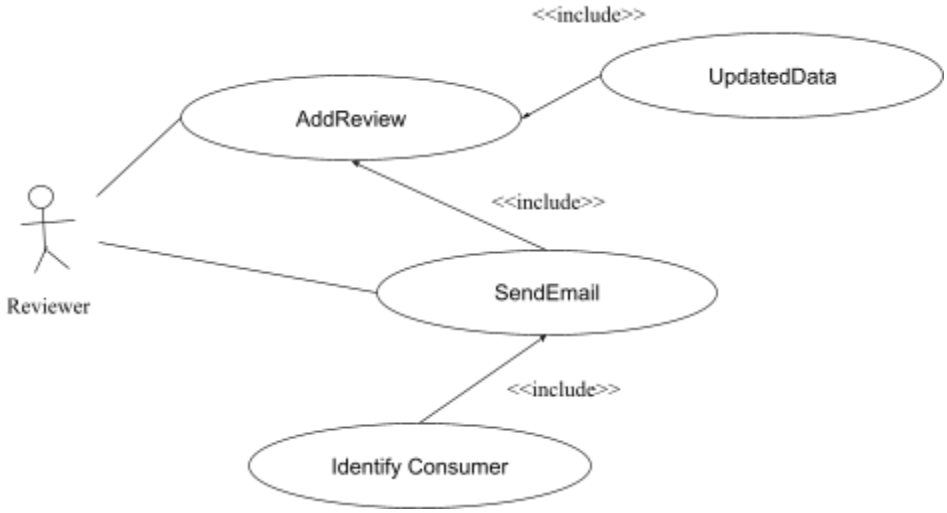
3.3.1 TicketBuyer



3.3.2 Administrator



3.3.3 Reviewer



3.4 Cinema Information

3.4.1 Tickets

3.4.1.1 Attributes

ticketID: Unique identifier for each ticket.

showtimeID: Identifier for the showtime the ticket is for.

seatNumber: Seat assignment for the ticket.

price: Cost of the ticket.

3.4.1.2 Functions

PurchaseTicket(): Marks the ticket as sold and processes payment.

CancelReservation(): Reverts a reserved ticket to available status.

ApplyDiscount(discountType): Applies a specified discount to the ticket price.

(Supports requirements for managing ticket sales, discounts, and reservations.)

UploadEmail(): Inputs the user's email address in the database for future use.

3.4.2 Showtimes

3.4.2.1 Attributes

userID: Takes email to identify user

showtimeID: Unique identifier for each showtime.

movieID: Identifier for the movie being shown.

theaterID: Identifier for the theater where the showtime takes place.

startTime: Start time of the showtime.

endTime: Calculated end time based on movie duration.

availability: Number of tickets available for the showtime.

3.4.2.2 Functions

CheckAvailability(): Checks how many tickets are still available for purchase.

UpdateAvailability(): Updates the ticket availability for the showtime.

GetShowtimeDetails(): Returns details about the showtime, including movie, theater, and timing.

(Facilitates the display of showtimes and their availability, critical for the ticket purchasing process)

AddReview(): Gives the option to add a review to the online page

3.5 Non-Functional Requirements

3.5.1 Ticket Updates

This system shall respond to user input within 2 seconds in usual conditions.

This system shall support real-time updates to ticket availability without impacting user experience.

This system shall process transactions within 5 seconds even during peak hours.

This system shall send an email to someone who has watched the movie and has an email in the

3.5.2 System Dependency

This system shall ensure data accuracy with a 99.99% success rate in transaction processing.

This system shall automatically recover from minor failures within 1 minute.

3.5.3 Seat Availability

This system shall be available 24/7, with scheduled maintenance windows communicated in advance.

This system shall provide failover capabilities to minimize downtime in the event of a server failure.

3.5.4 Cyber Security

This system shall implement encryption for all data transmission to protect user information.

This system shall adhere to industry-standard practices for data storage and password management.

This system shall have a dual-factor authentication for logging into the site.

This system shall conduct regular security audits to identify and mitigate potential vulnerabilities.

3.5.5 Maintainability

This system shall allow for updates and patches to be applied with minimal system downtime.

This system shall provide detailed logs to facilitate quick diagnosis and resolution of issues.

The system shall keep updated documentation on the API, codebase, and system architecture for further maintenance and development.

3.5.6 Portability

This system shall be compatible with major web browsers, including Chrome, Firefox, Safari, and Edge.

This system shall ensure responsive design for access via desktop.

This system shall allow for easy migration of the system database to different hosting environments if necessary.

3.6 Inverse Requirements

1. This system shall limit the number of tickets available for high-demand movies per user to prevent hoarding or scalping.
2. This system shall release any unclaimed tickets for resale to prevent loss of revenue due to empty seats.
3. This system shall ensure that the regular ticket prices remain competitive with other theaters to attract customers.
4. This system shall allow for cash payments at physical kiosks for users who prefer traditional payment methods.
5. This system shall periodically purge inactive or outdated email addresses to maintain data integrity and comply with privacy regulations.

3.7 Design Constraints

1. This software shall adhere to industry standards such as PCI-DSS for payment processing and GDPR for data protection to ensure legal compliance and user trust.
2. This system shall accommodate various hardware configurations, ensuring compatibility with different devices and screen sizes to provide a seamless user experience across platforms.
3. This system shall be scalable to handle increasing user traffic and ticket sales during peak periods without sacrificing performance or reliability.
4. This system shall be modular, allowing for easy integration with existing theater management systems and future expansion with new features or services.

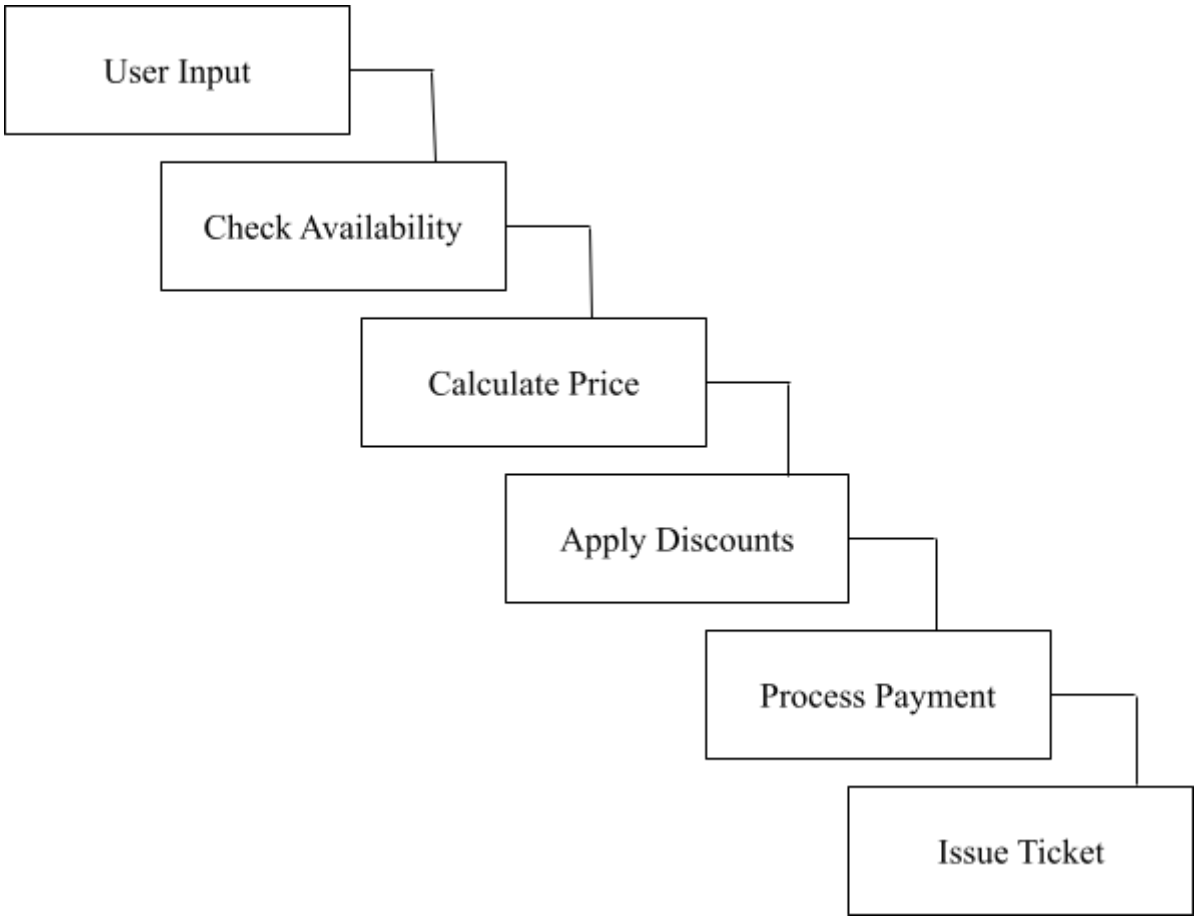
3.8 Logical Database Requirements

1. This system shall support various data formats for storing information such as user details, movie schedules, ticket availability, and transaction records.
2. This system shall have sufficient storage capacity to accommodate large volumes of data generated by ticket sales, customer interactions, and system logs.
3. This system shall enforce data integrity constraints to maintain the accuracy and consistency of stored information, preventing errors or corruption. Data
4. This system shall define policies for data retention to determine how long different types of data will be stored in the database before archiving or deletion.

3.9 Other Requirements

1. This system shall provide easy access to customer support channels, including FAQs, live chat, and email support, to assist users with inquiries or issues.
2. This system shall Implement a feedback mechanism for users to provide suggestions, report problems, or rate their experience, helping improve system usability and customer satisfaction.
3. This system shall set up performance monitoring tools to track system uptime, response times, and error rates, enabling proactive maintenance and troubleshooting.
4. This system shall conduct thorough integration testing to ensure seamless interaction between different system components, including user interfaces, payment gateways, and database operations.
5. This system shall develop comprehensive training materials and documentation for theater staff and administrators to onboard them smoothly onto the new ticketing system.

4. Analysis Models



4.1 Sequence Diagrams

Sequence diagrams will illustrate the flow of events between the ticket buyer, the system, and external interfaces. This includes payment processing and email services. For instance, a sequence diagram can outline the steps involved when a user selects a movie, chooses seats, applies discounts, and completes the purchase. It will also depict interactions with external systems such as payment gateways to process transactions and email services to send tickets to customers.

Each step in the process, from selecting a movie to receiving the ticket will be represented by a lifeline in the sequence diagram. It will focus on showing the order of interactions and any conditional branches based on user actions or system responses. These sequence diagrams will be closely tied to the functional requirements outlined in section 3.2.

4.2 Data Flow Diagrams (DFD)

Data flow diagrams will depict how data flows through the system from the initial input of user interactions to the final output of ticket purchases and reservations. They will identify the processes involved in handling user requests, updating ticket availability, processing payments, and sending tickets to customers.

For example, a data flow diagram can show how user inputs such as movie selections and seat choices are processed by the system to check ticket availability, apply discounts, calculate prices, and generate receipts. It can also illustrate how data is stored in databases and retrieved for reporting purposes.

These data flow diagrams will provide a clear visualization of the data flow and processing steps within the ticketing system. This will then aid in system design, implementation, and maintenance. Each of these analysis models will be developed in conjunction with the specific requirements outlined previously, ensuring that they accurately capture the behavior and functionality of the Movie Theater Ticketing System.

4.3 State-Transition Diagrams (STD)

In the Movie Theater Ticketing System, state-transition diagrams will represent the different states that a user session can be in, such as browsing movies, selecting seats, and purchasing a ticket. They will also illustrate the states of tickets, including availability, reserved, and sold. It then also shows how they transition between these states in response to user actions and system events.

For example, a state-transition diagram can show how a user session transitions from the initial state of browsing movies to selecting seats, applying discounts, and completing the purchase. Similarly, it can also then depict how ticket availability updates based on reservations and sales. These state-transition diagrams will help visualize the flow of states and transitions within the system, helping in understanding its behavior and ensuring that all possible states and transitions are accounted for in the system design.

Appendix

User Manual:

Welcome to the Movie Theater Ticketing System user manual. This document is designed to guide you through the process of using our online platform to browse, select, and purchase movie tickets. Whether you're a frequent moviegoer or a first-time user, this manual will help you navigate through the system effortlessly.

Logging In:

To access the Movie Theater Ticketing System, you need to log in with your credentials. If you don't have an account, you can sign up by clicking on the sign up button.

Browsing Movies:

Once logged in, you'll be directed to the homepage where you can browse through the list of available movies. You can filter movies based on genres, ratings, and showtimes to find the right movie for you.

Selecting a Movie:

After selecting a movie, you can view more details about it, including a brief description, ratings, and available showtimes. Choose the showtime that suits you best.

Reserving Seats:

Next, you'll be prompted to select your seats from the theater layout. The system will display the available seats in real-time, allowing you to choose your preferred seating arrangement.

Paying for Tickets:

Once you've selected your seats, proceed to the payment page. Here, you can choose from various payment options, including credit card, PayPal, or Bitcoin. Enter your payment details securely to complete the transaction.

Additional Features:

1. Viewing Past Interactions: You can access your purchase history and view details of past movies you've interacted with.
2. Movie Ratings: The system provides movie ratings and reviews from critics to help you make informed decisions.
3. Discounts and Promotions: Take advantage of discount codes and promotional offers during checkout to save on ticket prices.
4. Email Notifications: You'll receive email notifications with your ticket details for easy access.

Design Documents:

Verification Method:

The design implementation will be verified through a series of tests and reviews. Unit tests will be conducted to ensure that individual components function correctly, while integration tests will be used in order to validate the interaction between different modules of the system. User acceptance testing will be performed to assess the system's usability and functionality from the client's perspective.

Planned Modifications:

1. Enhanced User Interface: Continuous improvements to the user interface based on user feedback to enhance user experience.
2. Integration with Additional Payment Options: Integration with emerging payment options to provide more flexibility for users.
3. Expansion to Other Platforms: Expansion of the system to other platforms such as mobile applications to reach a wider audience.

Potential Future Modifications:

1. Integration with Loyalty Programs: Integration with theater loyalty programs to reward frequent customers.
2. Dynamic Pricing Strategies: Implementation of dynamic pricing strategies based on demand and other factors.
3. Enhanced Security Features: Continuous improvement of security features to mitigate potential threats and vulnerabilities.
4. Threat Identification and Countermeasures
5. Threats/System Vulnerabilities (Based on CIA Categorization)
6. Countermeasure: Implement strong encryption methods to protect sensitive data. Enforce strict access controls and authentication mechanisms to prevent unauthorized access.
7. Countermeasure: Implement data validation checks and integrity controls to detect and prevent unauthorized modifications to data.

Life-Cycle Model:

Phases:

1. Planning: Detailed analysis of the requirements outlined in the SRS and SDS, breaking them down into manageable user stories and tasks.
2. Development: Iterative development sprints focusing on implementing specific features or functionalities outlined in the SRS and SDS.
3. Testing: Continuous testing throughout development to ensure each feature meets the specified requirements and functions as expected.
4. Deployment: Incremental deployment of features as they are developed and tested, leading to a phased release of the complete system.
5. Feedback and Iteration: Gathering feedback from users and stakeholders after each deployment, incorporating necessary changes or enhancements into subsequent development iterations.

Models Used:

1. Scrum: Utilizing Scrum methodology for project management, including sprint planning, daily stand-ups, sprint reviews, and retrospectives.
2. Kanban: Implementing Kanban boards to visualize and track the progress of tasks throughout the development process, ensuring smooth workflow and efficient task management.
3. Continuous Integration/Continuous Deployment (CI/CD): Implementing CI/CD pipelines to automate the build, testing, and deployment processes, facilitating rapid and reliable delivery of updates to the production environment.

Reflection:

1. Effectiveness: Agile development allows for flexibility and adaptability, enabling the team to respond quickly to changing requirements or priorities outlined in the SRS and SDS.
2. Collaboration: The iterative nature of Agile promotes collaboration between cross-functional teams, fostering communication and synergy throughout the development process.
3. Feedback Loop: Regular feedback loops with stakeholders ensure that the final product aligns with their expectations and requirements, reducing the risk of misunderstandings or misinterpretations.
4. Risk Mitigation: By breaking down the development process into smaller, manageable iterations, Agile mitigates the risk of project failure or significant deviations from the initial SRS and SDS.

Challenges and Scalability:

1. Scalability: While Agile is well-suited for small to medium-sized teams and projects, scaling Agile practices to larger teams or more complex projects may pose challenges. However, implementing frameworks such as Scaled Agile Framework (SAFe) or Large-Scale Scrum (LeSS) can address scalability concerns.
2. Documentation: Agile prioritizes working software over comprehensive documentation, which may lead to potential challenges in maintaining thorough documentation throughout the development process. However, maintaining lightweight documentation and leveraging tools for documentation automation can help address this challenge.

Modifications and Improvements:

1. Refinement of SRS and SDS: Continuous refinement of the SRS and SDS based on feedback from stakeholders and evolving project requirements.
2. Enhanced Collaboration: Further enhancing collaboration between development teams, stakeholders, and end-users to ensure a shared understanding of project goals and priorities.
3. Integration of DevOps Practices: Integration of DevOps practices to streamline development, testing, and deployment processes, further enhancing the efficiency and reliability of software delivery.

Summary:

The Movie Theater Ticketing System aims to revolutionize the movie-going experience by providing a comprehensive online platform for ticket sales, scheduling, and customer management. The system targets both theater staff and patrons, offering an intuitive interface for seamless interactions.

The project's scope encompasses online ticket sales, distribution, and marketing, focusing on delivering convenience and accessibility to users. Stakeholders envision a system that efficiently manages showtimes, ticket availability, and user transactions while ensuring consistency and security across various theaters.

To meet these objectives, the system functions as a standalone interface with existing theater infrastructure to facilitate ticket sales and customer management. Key features include ticket sales, performance scheduling, customer management, and reporting capabilities, all aimed at enhancing the overall movie-going experience.

User characteristics vary between theater staff, who require efficient tools for managing performances and sales, and customers seeking seamless browsing, purchasing, and ticket retrieval experiences. The system must cater to diverse user preferences, ranging from online transactions to traditional methods of engagement.

Constraints such as technology compatibility and budget considerations shape system design and development. Assumptions regarding hardware, software, internet connectivity, and regulatory compliance guide the project's implementation.

Specific requirements detail external interfaces, functional capabilities, use cases, and non-functional aspects like ticket updates, system dependency, and cybersecurity measures. Analysis models, including sequence diagrams, data flow diagrams, and state-transition diagrams, visualize system behavior and data flow, aiding in system design and implementation.

In addition to its functional aspects, the project places significant emphasis on non-functional requirements such as system responsiveness, data integrity, and maintainability. The system is designed to handle high volumes of user interactions while ensuring real-time updates to ticket availability and processing transactions swiftly, even during peak hours. Cybersecurity measures, including encryption and dual-factor authentication, are implemented to protect user information and instill trust in the system. The system is also built with scalability and maintainability in mind, allowing for seamless updates, patches, and future expansions while ensuring minimal downtime and robust performance. Through meticulous attention to both functional and non-functional requirements, the Movie Theater Ticketing System aims to set a new standard for convenience, reliability, and security in the entertainment industry.

Conclusion:

The Movie Theater Ticketing System Software Requirements Specification presents a comprehensive overview of the proposed software solution. This document has undergone several revisions to ensure accuracy, completeness, and alignment with the stakeholders' requirements and expectations.

The system has an outline for the purpose and scope of the project, emphasizing the need for a user-friendly, efficient, and secure online ticketing system for movie theaters. It describes the general description of the product, including its functions, user characteristics, constraints, assumptions, and dependencies.

Also, the system then provides specific requirements, including external interface requirements, functional requirements, and non-functional requirements, catering to both administrators and customers. It details the hardware and software interfaces, user interfaces, functional processes, and error handling mechanisms necessary for the smooth operation of the system.

After, the system shows an analysis models such as sequence diagrams, data flow diagrams, and state-transition diagrams are included to visualize the system's behavior and functionality comprehensively. These models aid in system design, implementation, and maintenance, ensuring that all aspects of the Movie Theater Ticketing System are adequately addressed.

Additionally, the change management process is outlined to accommodate any future updates or modifications to the system, ensuring its relevance and applicability throughout the project lifecycle. Appendices provide supplementary information to enhance understanding and support project development.

In addition to serving as a blueprint for software development, the Movie Theater Ticketing System fosters clear communication and alignment among project stakeholders. By documenting requirements, dependencies, and constraints in detail, it ensures that all parties involved have a shared understanding of the project's objectives and deliverables. This clarity facilitates effective collaboration, reduces the risk of misunderstandings or discrepancies, and ultimately enhances the likelihood of project success.

Overall, the Movie Theater Ticketing System serves as a foundational document that guides the development process, aligning stakeholders' needs with the software solution. It sets the stage for the creation of a robust, user-centric, and efficient ticketing system that enhances the movie-going experience for both theater staff and patrons.