Theater Ticketing System

Software Requirements Specification

Version 1.3

March 27th, 2024

Group #17

Hosna Hyat
Isabelle Yuson
Edith Urquijo

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

Revision History

Date	Description	Author	Comments
2/7/2024	Version 1	Edith Urquijo	First Revision
2/14/2024	Version 1	Hosna Hyat	First Revision
2/14/2024	Version 1	Isabelle Yuson	First Revision
2/28/2024	Version 1.1	Edith Urquijo	Section 4 completion
2/28/2024	Version 1.1	Hosna Hyat	Section 4 completion
2/28/2024	Version 1.1	Isabelle Yuson	Section 4 completion
3/13/2024	Version 1.2	Edith Urquijo	Section 6 completion
3/13/2024	Version 1.2	Hosna Hyat	Section 6 completion
3/13/2024	Version 1.2	Isabelle Yuson	Section 6 completion
3/27/2024	Version 1.3	Edith Urquijo	Section 7 completion
3/27/2024	Version 1.3	Hosna Hyat	Section 7 completion
3/27/2024	Version 1.3	Isabelle Yuson	Section 7 completion

Document Approval

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

Signature	Printed Name	Title	Date
	<Your Name>	Software Eng.	
	Dr. Gus Hanna	Instructor, CS 250	

Table of Contents

1. Introduction

This SRS document will be used as a guideline to build a theater ticketing system. It will include the purpose of the software, a general description of the product, it will outline specific requirements, and will also outline the functionalities of the system. The aim of this document is to analyze how users can interact with the website system with side functions for said user to make their experience using the website simpler and easier

1.1 Purpose

The purpose of this SRS document is to collect and outline the different functionalities that users will have when trying to buy movie ticket(s). It helps to predict and account for the different situations that users can encounter.

This document describes the targeted audience here which are young adults and older who are trying to buy tickets for themselves and/or others. It describes how our website's functionality should be simple and easy enough for anyone of any age to understand and use.

1.2 Scope

I. *The Movie Theater Ticketing System will provide a comprehensive set of functionalities to facilitate ticket purchasing, food ordering, language preferences, payment options, and customer feedback. It will ensure a seamless and convenient experience for clients while adhering to certain limitations such as not accepting foreign cash currency, checks, and avoiding over-booking of theater rooms. These specifications define the scope and capabilities of the software product within the movie theater environment.*

   The software aims to efficiently handle ticket purchases, inquiries, and food orders both in-person, through the app, and online. A key objective is to minimize system crashes to ensure uninterrupted ticketing services for customers. In-person, app, and online purchases and inquiries should be consistently updated in real-time to provide accurate information to customers. The primary goal is to enhance customer satisfaction by providing a seamless and reliable ticketing experience across all channels.

   The scope outlined in this application description aligns with the broader objectives and goals defined in the System Requirement Specification. The emphasis on efficiency, reliability, and customer satisfaction remains consistent across both documents. The application of the software specified for the movie theater ticketing system focuses on efficiently handling purchases, inquiries, and food orders across multiple channels while prioritizing system reliability and real-time updates. The scope of the software product is defined by its ability to consistently deliver a seamless and satisfying experience to customers, aligning with higher-level specifications and objectives outlined in the System Requirement Specification.

II. **The product will:**

- *Know the maximum capacity for each theater room.*
- *Clients should be able to purchase tickets online through the movie theater's website.*
- *Clients should be able to purchase tickets through the movie theater's app.*
- *Clients should be able to purchase tickets in-person through the movie theater's ticketing kiosk.*
- *Clients should be able to order food, snacks, and beverages through the movie theater's app.*

- *Clients should be able to order food, snacks, and beverages in person through the movie theater's food and snack kiosk.*
- *Client(s) should be able to change language preference through the app or kiosk*
- *Software should be able to categorize movies according to genres.*
- *Software should be able to categorize movies according to release dates.*
- *Software should be able to accept local cash currency.*
- *Software should be able to accept Apple pay.*
- *Software should be able to accept Google pay.*
- *Software should be able to accept VeriFone.*
- *Software should be able to accept Visa.*
- *Software should be able to accept MasterCard.*
- *Software should be able to accept American Express*
- *Software should be able to accept credit unions.*
- *The software should be able to accept foreign banks.*
- *Customers should be able to provide their feedback/experience though the app and website.*

III. The product should not:

- The system should not accept foreign cash currency.
- Should not accept checks.
- Should not over-book theater rooms.

1.3 Definitions, Acronyms, and Abbreviations

Definitions:

1. ***Movie Theater Ticketing System:*** *A software solution designed to facilitate the purchase of movie tickets, inquiries about showtimes, and ordering of food and beverages within a movie theater environment.*
2. ***In-person Purchase:*** *The act of buying movie tickets or ordering food and beverages directly at the movie theater's physical location.*
3. ***App Purchase:*** *The process of purchasing movie tickets or ordering food and beverages through the movie theater's mobile application.*
4. ***Online Purchase:*** *Buying movie tickets or ordering food and beverages through the movie theater's website or online platform.*
5. ***Real-time Updates:*** *Continuous and immediate updates to ticket availability, showtimes, seating arrangements, and food orders to reflect the most current information.*

Acronyms and Abbreviations:

- ***SRS:*** *Software Requirements Specification*
- ***PCI DSS:*** *Payment Card Industry Data Security Standard*
- ***GDPR:*** *General Data Protection Regulation*
- ***WCAG:*** *Web Content Accessibility Guidelines*

- ***API:*** *Application Programming Interface*
- ***POS:*** *Point of Sale*
- ***ISP:*** *Internet Service Provider*
- ***CRM:*** *Customer Relationship Management*

1.4 References

This subsection should:
(1) Provide a complete list of all documents referenced elsewhere in the SRS, or in a separate, specified document.
(2) Identify each document by title, report number - if applicable - date, and publishing organization.
(3) Specify the sources from which the references can be obtained.
This information may be provided by reference to an appendix or to another document.

✔ *No references as of right now*
✔ *Placeholder list*

1.5 Overview

The theater ticketing system contains an online program (website) that allows up to many people to access the website simultaneously and buy tickets for a movie of their choice. It is organized in a way so that the user can order and buy tickets ahead of time for said movie or shortly after the movie starts. When the user goes onto the page of the selected movie, they're given many options such as checking out reviews, leaving customer feedback, or the possibility of leaving a discount to encourage the user to buy tickets

2. General Description

This software is required to handle multiple clients from an array of interfaces such as app, online or in-person, accept multiple forms of payments, distinguish which form(s) of payment(s) is acceptable, have a constant update of each theater rooms' capacity to not over-book tickets, and categorize movies based on genre.

2.1 Product Perspective

1. ***Client Interface Handling***

- *The software must support multiple client interfaces including:*
  a. *Mobile application (app)*
  b. *Online web interface*
  c. *In-person ticketing kiosks*

2. ***Payment Processing***

- *The system should be able to accept multiple forms of payments, including but not limited to:*
  a. *Credit/debit cards*
  b. *Mobile payment solutions (e.g., Apple Pay, Google Pay, Verifone)*
  c. *Cash (for in-person transactions)*
- *It should distinguish and handle each form of payment to ensure seamless transactions.*

3. ***Real-time Theater Room Capacity Updates***

- *The software must maintain a constant update of each theater room's capacity to prevent over-booking of tickets.*
- *It should dynamically adjust available seat counts as tickets are purchased in real-time.*

4. ***Movie Categorization***

- *The system should categorize movies based on genre to facilitate easier navigation and search for users.*
- *Genres may include but are not limited to: Action, Comedy, Drama, Horror, Science Fiction, Foreign, etc.*

5. ***Security***

- *The software must implement robust security measures to protect sensitive user data (e.g., payment information) and ensure compliance with relevant regulations.*

6. ***User Authentication and Authorization***

- *Implement authentication mechanisms to verify the identity of users accessing the system.*
- *Authorize users based on their roles (e.g., customer, theater staff, administrator) to control access to different functionalities.*

7. ***Reporting and Analytics***

- *Provide reporting capabilities to generate insights into ticket sales, revenue, and movie popularity.*
- *Analytics should aid in decision-making processes for theater management and marketing strategies.*

8. ***Integration***

- *The software should be capable of integrating with external systems such as:*
    a. *Theater management systems for real-time updates on showtimes and seating availability.*
    b. *Payment gateways for secure processing of transactions.*

9. Scalability and Performance

- *The system should be scalable to handle varying levels of user traffic, especially during peak movie times or promotional events.*
- *It should maintain optimal performance to ensure fast and responsive user experience across all interfaces.*

10. Accessibility

- *Ensure the software is accessible to users with disabilities, adhering to accessibility standards.*

11. Documentation and Support

- *Provide comprehensive documentation for system usage, administration, and troubleshooting.*
- *Offer user support channels (e.g., helpdesk, customer service phone number) for addressing user inquiries and issues.*

12. Regulatory Compliance

- *Ensure compliance with relevant industry regulations and standards, including but not limited to:*
    a. *Data protection laws*
    b. *Payment card industry standards*

13. User Experience (UX) Design

- *Design intuitive and user-friendly interfaces across all client channels for seamless navigation and ticket purchasing experience.*

14. Notifications and Alerts

- *Implement notifications and alerts to keep users informed about ticket availability, promotions, and upcoming movie releases.*

15. Localization and Internationalization

- *Support multiple languages and currencies to accommodate users from diverse regions.*

16. Backup and Disaster Recovery

- *Implement regular data backups and a disaster recovery plan to ensure data integrity and system resilience in case of unforeseen events.*

17. Technical Requirements

- *Specify the hardware and software infrastructure required to deploy and operate the system effectively.*

2.2 Product Functions

1. ***Ticket Booking:***

- *Allow users to browse available movies and showtimes.*
- *Enable users to select preferred movie, date, time, and seating options.*
- *Facilitate secure booking and reservation of tickets across multiple interfaces (app, online, in-person).*

2. ***Payment Processing:***

- *Accept multiple forms of payments including credit/debit cards (including foreign banks), mobile payments, and cash.*
- *Validate payment information and process transactions securely.*
- *Provide users with electronic or printed tickets upon successful payment.*

3. ***Real-time Theater Room Updates:***

- *Continuously monitor and update theater room capacities in real-time.*
- *Ensure that available seat counts are accurate to prevent overbooking.*

4. ***Movie Categorization and Search:***

- *Categorize movies based on genre, ratings, release date, and other relevant criteria.*
- *Implement a search functionality to allow users to easily find movies of interest.*

5. ***User Authentication and Authorization:***

- *Authenticate users to access booking and payment functionalities.*

- *Authorize users based on their roles (e.g., customer, theater staff, administrator) to control access to different features.*

6. ***Reporting and Analytics:***

- *Generate reports on ticket sales, revenue, and movie popularity.*
- *Provide analytics to aid in decision-making processes for theater management.*

7. ***Integration:***

- *Integrate with theater management systems to retrieve real-time showtime and seating information.*
- *Connect with payment gateways for secure processing of transactions.*

8. ***Notifications and Alerts:***

- *Send notifications and alerts to users regarding ticket availability, promotions, and upcoming movie releases.*

9. ***Accessibility:***

- *Ensure accessibility features for users with disabilities.*

10. ***Localization and Internationalization:***

- *Support multiple languages and currencies to accommodate users from diverse regions.*

11. ***Backup and Disaster Recovery:***

- *Implement regular data backups and a disaster recovery plan to ensure data integrity and system resilience.*

12. ***User Experience (UX) Design:***

- *Design intuitive and user-friendly interfaces across all client channels for seamless navigation and ticket purchasing experience for all ages and backgrounds.*

13. ***Security:***

- *Implement robust security measures to protect sensitive user data and ensure compliance with relevant regulations.*

14. ***Scalability and Performance:***

- *Ensure the system is scalable to handle varying levels of user traffic and maintain optimal performance.*

15. ***Documentation and Support:***

- *Provide comprehensive documentation for system usage, administration, and troubleshooting.*
- *Offer user support channels (e.g., helpdesk)for addressing user inquiries and issues.*

2.3 User Characteristics

If user is a bot:

- *They are blocked by the system (indicated by if bot is trying to buy many tickets at once to high demand movies*)

If the user is a real person but not an admin:

- *They have access to the "normal/usual" website.*
- *Here they can view what movies are out and see the corresponding seats and times available for that movie.*
- *They can also get discounts (this is in the case if we make it so there's a membership program or something + student and/or elderly discounts)*
- *Outside buying a ticket, the user also has access to the customer feedback system and trending reviews of said movie.*

If the user is a real person and an admin:

- *They have access to the administrator version/mode of the website.*
- *Admin can switch the view of the website from the perspective of a regular user (so they're able to see what the "normal/usual" website looks like)*
- *Admins have access to everything normal users have.*
- *Are able to issue and control refunds*
- *Are able to"kick" people off seats*
- *Can control how the layout is displayed (ex. choosing what reviews are shown specifically)*

2.4 General Constraints

1. ***Technology Stack:***

- *The system must be developed using specific technologies or programming languages already approved by the organization (e.g., Java, Python, C++).*

2. ***Compatibility:***

- *The software must be compatible with existing hardware and software infrastructure used by the theater chain (e.g., operating systems, databases, web browsers).*

3. ***Regulatory Compliance:***

- *The system must adhere to industry regulations and standards for secure payment processing and data protection.*

4. ***Budget Constraints:***

- *Development costs must stay within the allocated budget, limiting the selection of third-party tools or services that require additional licensing fees.*

5. ***Time Constraints:***

- *The system must be developed and deployed within a specific timeframe to align with marketing and operational schedules, limiting the scope of features and functionalities that can be implemented.*

6. ***Security Requirements:***

- *The system must meet strict security requirements.*

7. ***Scalability Requirements:***

- *The system must be designed to accommodate future growth and scalability needs.*

8. ***User Accessibility:***

- *The system must comply with accessibility standards to ensure equal access for users with disabilities.*

9. Branding Guidelines:

- *The system's user interface and branding must align with the theater's branding guidelines and visual identity. Design choices must relate to color schemes, and logos.*

10. Localization Requirements:

- *The system must support localization for multiple languages and regions, constraining design choices related to text expansion, layout adjustments, and cultural considerations.*

11. Data Privacy:

- *The system must adhere to strict data privacy policies, limiting data collection, storage, and processing options to ensure compliance with regulations.*

12. Maintenance and Support:

- *The system must be designed for ease of maintenance and support, imposing constraints on architectural choices and code complexity to facilitate ongoing updates and bug fixes.*

13. Integration Requirements:

- *The system must integrate seamlessly with third-party systems, payment gateways, and customer relationship management tools, limiting design choices that may hinder integration.*

14. User Experience Guidelines:

- *The system's user interface and user experience must meet predefined guidelines and usability standards, constraining design choices related to navigation, layout, and interaction patterns.*

15. Performance Requirements:

- *The system must meet specific performance requirements (e.g., response time, throughput) under expected load conditions, limiting design choices that may impact performance negatively.*

2.5 Assumptions and Dependencies

1. ***Availability of Theater Hardware and Infrastructure:***
    - *The assumption is that the movie theater's hardware infrastructure, including servers, networking equipment, and ticketing kiosks, is in place and operational.*
2. ***Stable Internet Connectivity**:*
    - *It is assumed that the movie theater premises have stable internet connectivity to support online ticket booking and payment processing without interruptions.*
3. ***Availability of Movie Showtime Data:***
    - *It is assumed that accurate and up-to-date movie showtime data, including schedules and seating availability, will be provided by the theater management system.*
4. ***Compliance with Regulatory Requirements:***
    - *It is assumed that the movie theater complies with relevant regulations and standards.*
5. ***User Accessibility:***
    - *It is assumed that the movie theater ticketing system will be accessible to users with disabilities.*
6. ***Availability of Payment Gateways:***
    - *It is assumed that the designated payment gateways for online transactions are available and integrated into the system for secure payment processing.*
7. ***Maintenance and Support:***
    - *It is assumed that adequate resources and support personnel will be available for ongoing maintenance and support of the movie theater ticketing system.*

Dependencies:

1. ***Third-Party APIs and Services:***
    - *The movie theater ticketing system depends on the availability and reliability of third-party APIs and services for functionalities such as movie showtime data, payment processing, and seat availability.*
2. ***Hardware Compatibility:***
    - *The system's compatibility with theater hardware and infrastructure, including ticketing kiosks and payment systems, is crucial for seamless integration and operation.*
3. ***Internet Service Third-Party APIs and Services:e Providers (ISPs):***
    - *The system's performance and reliability depend on the internet service providers (ISPs) serving the movie theater premises, as stable internet connectivity is essential for online booking and payment processing.*
4. ***Regulatory Compliance Updates:***
    - *Any changes or updates to regulatory requirements, such as new data protection laws or payment processing regulations, may impact the system's design and functionalities, requiring adjustments to ensure compliance.*
5. ***Changes in User Preferences:***

- *The system's requirements may be influenced by changes in user preferences and behavior, such as shifts towards mobile ticketing or new expectations for user experience.*

6. ***Theater Management System Integration:***
   - *Integration with the theater management system for real-time updates on showtimes, seating availability, and other relevant data is essential for the system's functionality.*
7. ***Payment Gateway Integration:***
   - *The system's integration with designated payment gateways for secure payment processing is a critical dependency for enabling online transactions.*

3. Specific Requirements

This section outlines the detailed requirements that guide the design, implementation, and testing of this product. Each requirement is carefully crafted to ensure correctness, traceability, unambiguity, verifiability, prioritization, completeness, consistency, and uniqueness.

3.1 External Interface Requirements

3.1.1 User Interfaces

The ticketing system will be available on any browser such as Safari, Google Chrome, Edge and so on.

3.1.2 Hardware Interfaces

The ticketing system will be accessible via web interface. All hardware will require access to the internet.

3.1.3 Software Interfaces

1. *The ticketing system shall communicate via web interface. This menu-driven interface will make it more user friendly and quicker to use.*
2. *The ticketing system shall communicate with the database that stores information. This is to keep track of and display what seats are taken for specific times and for what movies.*
3. *The ticketing system shall communicate with the given information of the user. This is to keep track of whether the user has a membership. This also keeps track of if a discount is necessary.*
4. *The ticketing system shall communicate with the money system and payment gateways. This is to be able to access and properly use the displayed payment methods.*

3.1.4 Communications Interfaces

The ticketing system will require the needed protocol for communication over the internet to establish communication between the user/admins and the website system.

Communication will involve the payment gateways, management of the theater system, database of the movies, database of customer feedback and reviews, and the database of the email system for admins.

3.2 Functional Requirements

This format outlines specific functional requirements or features, providing detailed information on inputs, processing, outputs, and error handling for each requirement. Further functional requirements or features can be listed following the same structure.

3.2.1 <Functional Requirement or Feature #1>

3.2.1.1 Introduction

- *This section provides an introduction to the functional requirement or feature.*
- ***Feature 1:** Selecting a movie and quantity of tickets.*

3.2.1.2 Inputs

- ***User Input:** Customer selects a movie and specifies the number of tickets required.*
- ***Payment Information:** User provides payment details such as credit/debit card information, mobile payment details, or cash.*

3.2.1.3 Processing

- ***Validation:** The system validates the selected movie and the number of tickets against available showtimes and seating capacity.*
- ***Payment Processing:** If payment is made online, the system processes the payment securely through integrated payment gateways. If payment is made in person, the system verifies and records the transaction using appropriate POS systems.*

3.2.1.4 Outputs

- ***Confirmation:** Upon successful processing, the system generates a booking confirmation with details such as movie title, showtime, seat numbers, and total amount paid.*
- ***Ticket:** The system generates electronic or physical tickets for the customer, depending on the preferred delivery method.*

3.2.1.5 Error Handling

- ***Invalid Selection**: If the selected movie or number of tickets is invalid or unavailable, the system prompts the user to make appropriate changes.*
- ***Payment Failure:** In case of payment failure, the system notifies the user and provides instructions to resolve the issue or select another payment method.*

3.2.2 <Functional Requirement or Feature #2>

3.2.2.1 Introduction

- *This section provides an introduction to the functional requirement or feature.*
- ***Feature 2:** Selecting specific date and time for specified movie.*
-

3.2.2.2 Inputs

- ***User Input:** Customer selects a movie and specifies the desired date and time for the show.*

3.2.2.3 Processing

- ***Showtime Availability:*** *The system receives available showtimes for the selected movie and date.*
- ***Seating Availability:*** *The system checks the availability of seats for the selected showtime.*

3.2.2.4 Outputs

- ***Available Showtimes:*** *The system presents a list of available showtimes for the selected movie and date.*
- ***Seating Availability:*** *The system displays the number of available seats for each showtime.*

3.2.2.5 Error Handling

- ***No Showtimes Available:*** *If no showtimes are available for the selected movie and date, the system informs the user and suggests alternative options.*
- ***Insufficient Seating:*** *If there are insufficient seats available for the selected showtime, the system prompts the user to select a different showtime or movie.*

3.3 Use Cases

3.3.1 Website Interface

1. *The ticketing system will have icons by which the user can click on to select a movie on the landing page and the time and date on a subsequent page.*

2. *The ticketing system will allow users to log into their account via a sign-in box if they are movie members.*

3. *The ticketing system will showcase movies that will be released within the next month.*

4. *The ticketing system will suggest which movie to watch based on positive movie reviews.*

3.3.2 Movie members

1. *The ticketing system will allow movie members to choose their seat in the theater.*

2. *The ticketing system will allow movie members to receive a discount at the payment window.*

3. *The ticketing system will reward movie members with points to redeem upon their next return.*

4. *The ticketing system will allow movie members to reset their password via email if their number of password attempts has been exceeded.*

3.3.3 Unauthorized users

1. *The ticketing system will end the purchasing session if captcha requirements are not met therefore deeming the user a bot.*

2. *The ticketing system will end the purchasing session if a payment is not made within 15 minutes of choosing the movie, date, and time.*

3. *The ticketing session will end the purchasing session if a user attempts to buy more than 20 tickets at a time.*

4. *The ticketing system will end a purchasing session if the card input is incorrect.*

3.4 Classes / Objects

This section describes the classes or objects present in the software project.

3.4.1 <Class / Object #1>

3.4.1.1 Attributes

- *Name: Ticket*
- *Description: Represents a ticket for a movie.*
- *Attributes:*
  Movie Title
  Showtime
  Seat Number
  Ticket Price
  Ticket Status (e.g., booked, canceled)

3.4.1.2 Functions

- *Functions:*
  reserveTicket(): Reserves a ticket for a specific showtime and seat.
  cancelTicket(): Cancels a previously reserved ticket.
  getTicketDetails(): Retrieves details of a specific ticket.

Reference to Functional Requirements and/or Use Cases

- *Functional Requirement 1: Handle ticket reservations and cancellations.*
- *Functional Requirement 2: Provide details of purchased tickets.*
- *Use Case: Customer reserves and cancels tickets through the ticketing system.*

3.4.2 <Class / Object #2>

3.4.2.1 Attributes

- *Name: Movie*
- *Description: Represents a movie available for screening at the theater.*
- *Attributes:*
  Title
  Genre
  Release Date
  Duration
  Director
  Cast
  Synopsis

3.4.2.2 Functions

- *Functions:*
  - *getMovieDetails(): Retrieves details of a specific movie.*
  - *checkShowtimes(): Retrieves available showtimes for the movie.*
  - *checkAvailability(): Checks the availability of seats for a specific showtime.*

Reference to Functional Requirements and/or Use Cases

- *Functional Requirement 2: Provide information on available showtimes for movies.*
- *Functional Requirement 3: Categorize movies based on genre and release date.*
- *Use Case: Customer views movie details and available showtimes through the ticketing system.*

3.4.3 <Class / Object #3>

3.4.3.1 Attributes

- *Name: FeedbackSystem*
- *Description: Class that acts as a database for reviews and customer feedbacks*
- *Attributes:*
  - *Review*
  - *Rating*
  - *Feedback*

3.4.3.2 Functions

- *Functions:*
  - *createReview(): Adds reviews given by the user onto an existing list*
  - *displayReview(reviewNum): Displays the called review*
  - *getRating(): Adds rating (0 - 5) given by the user onto its corresponding review*
  - *createFeedback(): Adds feedback given by the user onto an existing list*

Reference to Functional Requirements and/or Use Cases

- *Functional Requirement 1/4: Needs an existing list to store reviews*
- *Functional Requirement 2: Needs an existing display for the admin to choose what reviews are on the website's layout*
- *Functional Requirement 4: Feedback is sent to the admin and is hidden from normal users once submitted*
- *Use Case: Customer can view the current reviews and/or send feedback before, during, or after the process of trying to buy a ticket to help them make a decision or if the user runs into problems*

3.5 Non-Functional Requirements

3.5.1 Performance

The ticketing system will be web based and can be accessed by any user on a computer or mobile device. How fast the interface loads as well as how fast the purchasing page loads is fully dependent on the strength of the user's internet connection and their device capabilities.

3.5.2 Reliability

The user can ensure that the web interface will be fully functional at any time of day. If the user needs to purchase movie tickets days in advance, the option is there. If the movie starts within a minute the user will still be allowed to make a purchase to attend the viewing.

3.5.3 Availability

The ticketing system will be available 95% of the time. Service updates will be performed at a non-peak hour to ensure users are not interrupted at the time of making a purchase.

3.5.4 Security

Movie members who choose to login to their account to receive member perks will be asked if they would like the system to remember their password. If "no" is selected, members will be required to provide their password every time they wish to make a purchase. If "yes" is selected, the user will be prompted to set up a single factor authentication for added security.

3.5.5 Maintainability

The ticketing system will be updated once a week based on customer reviews and website status reports. If there are too many inconsistencies in loading times then we can assume it is not all due to different user devices and a fix must be made.

3.5.6 Portability

This ticketing system is portable and can be used on other computer environments. The build of this ticketing system is manipulable and can be altered to fit other environment needs with ease.

3.6 Inverse Requirements

A user can request to use a phone number instead of an email address to login to their movie member account.

A user can request for a discount on any and all movies before 2 pm on a weekday regardless if you are a movie member or not.

3.7 Design Constraints

One of the design constraints is that the web interface cannot account for every possible error.

3.8 Logical Database Requirements

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.

3.9 Other Requirements

Catchall section for any additional requirements.

4. Analysis Models

In the development of specific requirements in the Software Requirements Specification (SRS) for the Theater Ticketing System, the following analysis models are utilized.

4.1 Architectural Diagram of All Major Components

```
+-----------------+
|  User           |
|  Interface      |
+--------+--------+
         |
         | HTTP/HTTPS
         |
+--------v--------+      +-----------------+
| Ticketing       |      |   Payment       |
|  System         |      |  Gateway        |
+--------+--------+      +-----------------+
         |                  |
         |                  |
         |                  |
         +------------------------+
                  RESTful APIs
         +------------------------+
         |
         |
         |
         v
+-----------------+
|  Theater        |
|  Management          |
|  System         |
+-----------------+
```

1. *User Interface:*

· Represents the interface through which users interact with the ticketing system.

· This component is responsible for receiving user input such as movie selections, ticket quantities, and payment details.

2. *Ticketing System:*

· Acts as the core of the ticketing application, responsible for managing ticket reservations, cancellations, and payment processing.

· Receives requests from the User Interface and communicates with external systems.

3. *Payment Gateway:*

· Handles payment processing for ticket purchases made by users.

· Receives payment information from the Ticketing System and communicates with external payment processing services or financial institutions to authorize and process payments.

4. *Theater Management System:*

· Represents the backend system responsible for managing theater operations, including showtimes, seating arrangements, and inventory management.

· The Ticketing System interacts with the Theater Management System to retrieve real-time information about available showtimes, seating availability, and other relevant theater data.

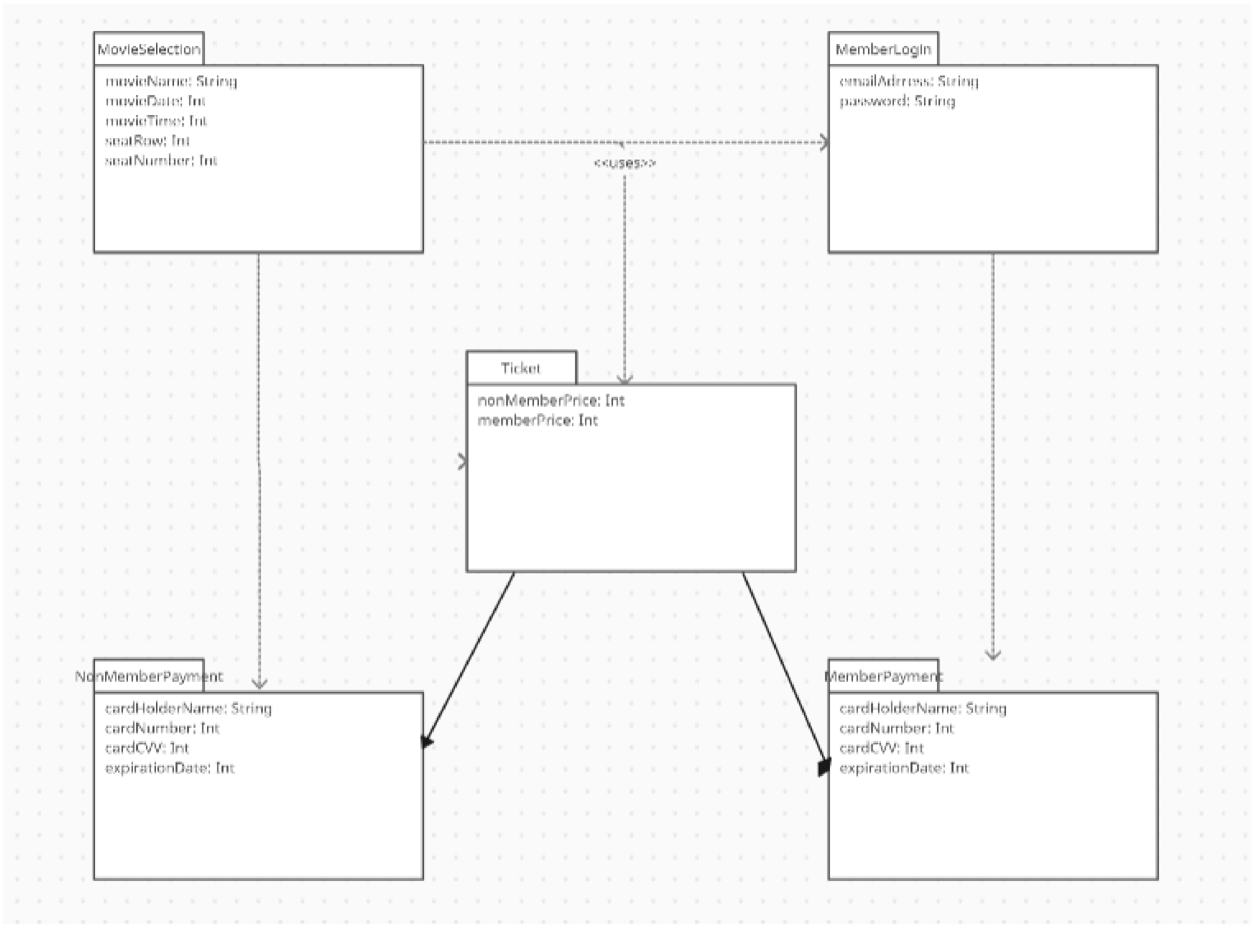
5. *Interactions:*

· The User Interface communicates with the Ticketing System using protocols, showing the flow of user requests and responses.

· The Ticketing System communicates with the Payment Gateway, enabling secure payment processing and transaction management.

· Additionally, the Ticketing System interacts with the Theater Management System to retrieve theater-related information, facilitating accurate ticket reservations and showtime availability updates.

4.2 UML Class Diagram



4.3 Description of classes

- *The MovieTicketingSystem class provides the following methods: movieSelection, movieLogin, ticket, nonMemberPayment, memberPayment.*
- *In the movieSelection method we are declaring the strings and ints that are associated with the type of movie that will be selected by the user.*
- *The movieLogin method declares strings for the user to login to their account in order to access their membership deals such as discounts. This method will take inputs and output "wrong username/password" if user inputs are incorrect. It will otherwise let the user login. If-else statements will be used.*
- *The ticket method will provide the user with ticket pricing depending on whether they are a member or not. If-else statements will also be used to determine which conditions are met.*
- *The nonMember class will allow a user who does not have a movie membership to make their payment without any discounts applied.*
- *The member class will allow a user who does have a movie membership to make their payment with discounts applied.*

4.4 Description of attributes

- *MovieSelection*
  - *movieName: name of the movie*
  - *movieDate: the date selected to watch the movie on*

- *movieTime: the time the movie is showing*
    - *seatRow: the corresponding row of where the seat is*
    - *seatNumber: the number of the seat selected*
- *NonMemberPayment*
    - *cardHolderName: the name of the user who's paying*
    - *cardNumber: the number of the card that the user is using to pay*
    - *cardCVV: the CVV of the user's card*
    - *expirationDate: the expiration date of the users card*
- *Ticket*
    - *nonMemberPrice: the listed price of the ticket for nonmembers*
    - *memberPrice: the listed price of the ticket for members*
- *MemberLogin*
    - *emailAddress: the email address of the user*
    - *password: the password of the user's membership account*
- *MemberPayment*
    - *cardHolderName: the name of the user who's paying*
    - *cardNumber: the number of the card that the user is using to pay*
    - *cardCVV: the CVV of the user's card*
    - *expirationDate: the expiration date of the users card*

4.5 Description of operations

- *reserveTicket( )*
    - *reserves a ticket for a specific showtime and seat*
- *cancelTicket()*
    - *cancels a reserved ticket (ticket must be under the reserved status)*
- *getTicketDetails*
    - *getter function that displays details of the ticket*
- *getMovieDetails()*
    - *getter function that displays the details of the movie*
- *checkShowtime()*
    - *retrieves and displays the showtimes for a movie*
- *checkAvailability()*
    - *checks the availability of seats for a specific showcase*
- *createReview()*
    - *adds reviews given by the user (or put it by the admin) onto an existing list*
- *displayReview()*
    - *displays a specific review from the list on the website*
- *getRating()*
    - *getter function that gets the rating imputed from the user (from reviews)*
- *createFeedback()*
    - *adds feedback given by the user onto an existing list*

4.6 Development plan and timeline

- *Isabelle - formatting and GitHub management*
- *Edith - UML diagram + UML description + code management*
- *Hosna - SWA diagram + SWA description + code management*

5. Change Management Process

Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.

6. Test Plan Outline

Refer to attached: https://1drv.ms/x/s!AmmPrhowW8Dltkf0GS-1Y017ryTK?e=Me4tkC

- *Server/browser Integrity*
    1. *A Google search to test the landing ticketing page.*
        - *server is fast and responsive*
        - *cover for failed layout or broken links*
    2. *A browser compatibility test to ensure the website can be accessed from any internet site.*
        - *can respond effectively based on device being used (phone, computer, etc)*
- *User Interface*
    1. *A membership registration/login test to make sure the user is logged in if the correct criteria is met.*
    2. *A ticket reservation test to make sure the website handles the reservations correctly and executes the correct output.*
    3. *A ticket cancellation test to make sure the system executes a refund after the cancellation is confirmed.*
    4. *A showtime availability test to make sure the software system correctly displays the amount of open seats versus what has already been purchased.*
        - *booking tickets for available seats*
        - *attempting to book tickets for already booked seats*
        - *ability to cancel reservations*
        - *verifying seat availability*
    5. *A user interface test to ensure the website correctly navigates the user from one page to another depending on the options selected.*
- *Payment Processing*
    1. *A payment processing test to ensure payment goes through and tickets are issued afterwards.*
        - *confirmation of payment*
        - *successfulness of payment*
        - *handling different methods of payment*
        - *handing payment errors*
        - *handling of security breaches or unauthorized payments*
- *Performance Issues*
    1. *An error handling test for login credentials to ensure a user is not a bot and/or inputting incorrect credentials on the login page.*
        - *handling of user not using the site features correctly*
        - *handling of high user traffic*
    2. *An error handling test for payment failure to handle incorrect form of payment such as incorrect card name, card number, cvv number, or expiration date.*
- *Feedback database*
    1. *A feedback submission test to ensure that customer reviews are processed correctly and not being deleted or sent to an incorrect database.*
        - *handling of given feedback and storing it into the database*
        - *display of correct reviews/feedback*

7. Data Management Strategy

The data management strategy for the Theater Ticketing System is designed to guarantee efficient storage, retrieval, and security of persistent data while maintaining data integrity and consistency. By choosing an SQL-based database and following only the best practices in database design and security, the system can efficiently and effectively handle the various data requirements of the ticketing platform.

1. Database Technology:

- ***Relational Database Management System (RDBMS):*** *Utilizing an SQL-based RDBMS for data storage and management.*

2. Database Design:

- ***Single Database Approach:*** *All data related to the theater ticketing system are stored in a single database.*

3. Logical Data Organization:

- ***Normalized Schema:*** *Employing a normalized database schema to minimize data redundancy and guarantee data integrity.*
- ***Entity-Relationship Model (ER Model):*** *Designing the database schema using ER modeling techniques to represent the relationships between different data entities.*

4. Data Partitioning:

- ***Logical Data Partitioning:*** *Organizing data into separate tables based on logical entities such as users, movies, tickets, payments, and theaters.*
- ***Indexing:*** *Implementing appropriate indexing on key columns to optimize data retrieval performance.*

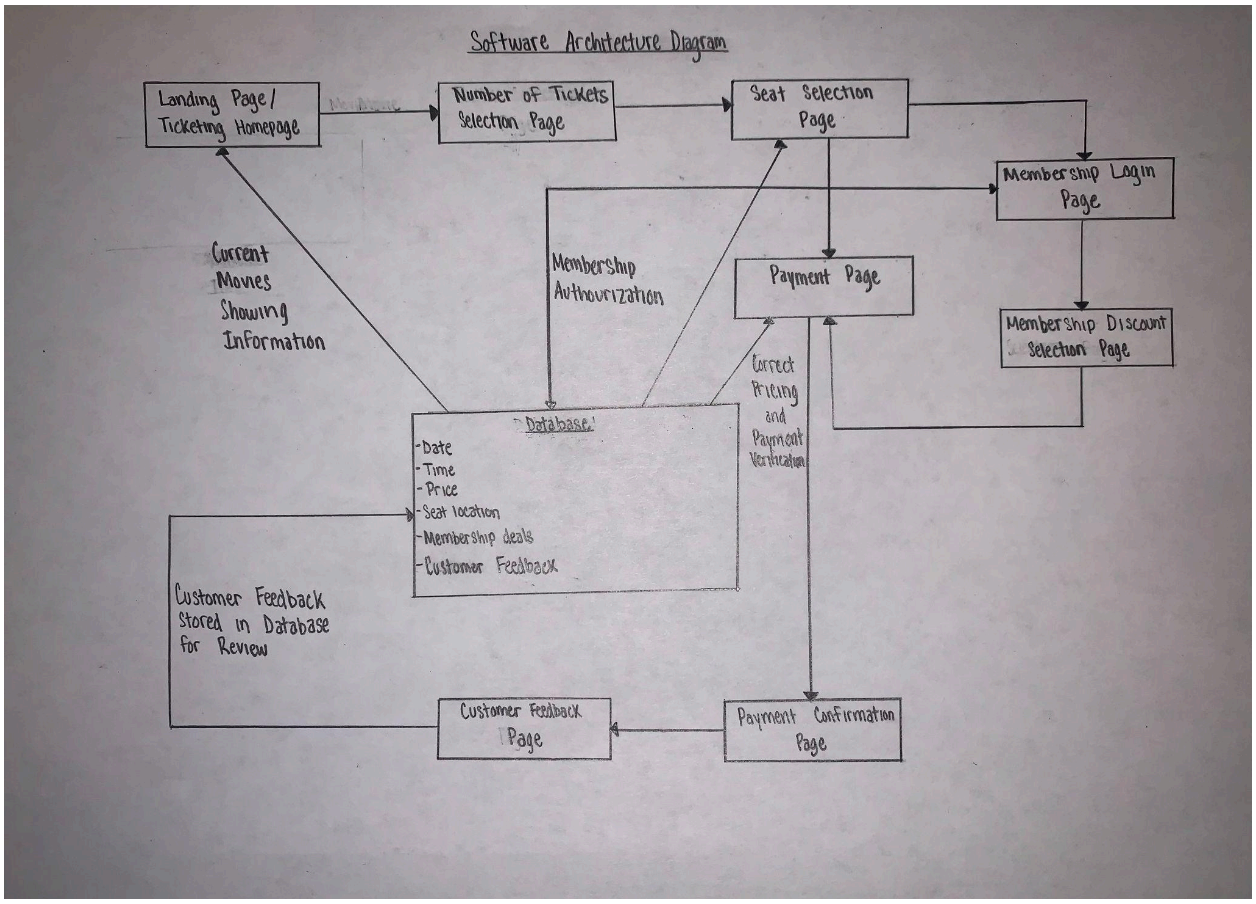
5. Data Security:

- ***Encryption:*** *Implementing data encryption techniques to secure sensitive information such as user information, and payment details.*
- ***Access Control:*** *Enforcing role-based access control (RBAC) to restrict access to sensitive data based on user roles and permissions.*

6. Backup and Disaster Recovery:

- ***Regular Backups:*** *Performing regular backups of the database to prevent data loss in case of system failures or disasters.*
- ***Redundancy:*** *Implementing database redundancy and failover mechanisms to ensure high availability and data reliance.*

7.1 Software Architecture Diagram


Software Architecture Diagram
Landing Page / Ticketing Homepage
Number of Tickets Selection Page
Seat Selection Page
Membership Login Page
Payment Page
Membership Discount Selection Page
Current Movies Showing Information
Membership Authourization
Correct Pricing and Payment Verification
Database:
-Date
-Time
-Price
-Seat location
-Membership deals
-Customer Feedback
Customer Feedback Stored in Database for Review
Customer Feedback Page
Payment Confirmation Page

7.2 SQL vs Non-SQL

Relational Database Management System (RDBMS):

- ***<u>Reasoning:</u>*** *This system is meant to interact with structured data that can be organized into tables with predefined relationships. An RDBMS offers strong "ACID" (Atomicity, Consistency, Isolation, and Durability) compliance, and support for complex queries and transactions, which are all essential characteristics needed for managing various aspects of ticket booking, user accounts, payments, and theater information.*
- ***<u>Benefits:</u>*** *With an RDBMS, this can ensure data integrity, enforce referential integrity constraints (ensuring the validity of references), and perform efficient data retrieval and manipulation using SQL queries. This choice supports the structured nature of the data and the relational dependencies between different entities that are presented in the theater ticketing system.*
- ***<u>Trade-offs Considered:</u>*** *While RDBMS solutions may have scalability limitations compared to non-SQL databases, the initial scope of the project and the emphasis on data integrity and consistency make the trade-offs acceptable. As the system scales, strategies such as database partitioning or replication can be implemented to address scalability concerns.*

Normalized Schema:

- ***Reasoning:*** *A normalized schema is chosen to minimize data redundancy, maintain data integrity, and facilitate efficient data updates and modifications. Given the diverse data entities such as users, movies, tickets, payments, and theaters, a normalized schema allows for better organization and management of the data.*
- ***Benefits:*** *Normalization reduces the risk of data anomalies such as update anomalies, insertion anomalies, and deletion anomalies by eliminating redundant data and maintaining dependencies through foreign key constraints. This ensures that the data remains consistent and accurate throughout its lifecycle.*
- ***Trade-offs Considered:*** *While normalization may lead to increased join operations and slower query performance for complex queries, the benefits of data integrity and flexibility in schema modifications substanctially outweigh potential performance impacts. Proper indexing and query optimization techniques can help ease performance issues as the system continues in growth.*

In summary, the Relational Database Management System with a normalized schema is a perfect fit for the Theater Ticketing System due to its suitability for managing structured data and enforcing data integrity constraints. This choice aligns well with the requirements of the system and ensures efficient storage, retrieval, and security of persistent data.

7.3 Design Decisions

Single Database:

- ***Single Database Approach:*** *Chosen for simplicity and ease of management. Reduces complexity in data synchronization and maintenance.*
- ***Trade-off:*** *May lead to performance issues as the system scales and data volume increases. It may also pose challenges in data isolation and security sectors.*

Data Security:

- ***Encryption and Access Control:*** *Prioritized for protecting sensitive data from unauthorized access and data breaches.*
- ***Trade-off:*** *Encryption and access control mechanisms may become an overhead in terms of computational resources and query execution time. However, the security of the user data is enormous, and the performance impact is acceptable compared to the benefits of enhanced data security.*

Relational Database:

- ***Relational Database (SQL):*** *Selected for its strong consistency, ACID (Atomicity, Consistency, Isolation, and Durability) compliance, and support for complex queries and transactions.*
- ***Trade-off:*** *May encounter scalability limitations compared to non-SQL databases. However, considering the initial scope of this project, consistency and reliability of an SQL database outweighs any current potential scalability concerns.*

Normalized Schema:

- ***Normalized Schema:*** *Selected for reducing data redundancy, maintaining data integrity, and facilitating data updates and modifications.*
- ***Trade-off:*** *May lead to increased joint operations and slower query performance, especially for complex queries involving multiple tables. However, the benefits of data integrity and flexibility in schema modifications outweigh potential performance impacts.*

A. Appendices

Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS's overall set of requirements.

Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.

A.1 Appendix 1

A.2 Appendix 2