Rex Movies

Software Requirements Specification

6.0

Feb. 14, 2024

Group 14

Skyler Varney, Jose Liborio, Carlos Venegas

Prepared for

CS 250- Introduction to Software Systems

Instructor: Gus Hanna, Ph.D.

Spring 2024

Revision History

Date	Description	Author	Comments
2/08/24	SRS 1.0	Skyler, Jose, Carlos	Rough ideas in.
2/11/24	SRS 2.0	Skyler, Jose, Carlos	Overall changes to SRS. More descriptive language.
2/13/24	SRS 3.0	Skyler, Jose, Carlos	Added Use Case Diagram. More detail.
2/28/24	SRS 4.0	Skyler, Jose, Carlos	Corrected gaps, formatting. Added UML Diagram, SWA Diagram, descriptions, etc.
3/14/24	SRS 5.0	Skyler, Jose, Carlos	Updated UML, and Classes/Objects. Added Verification Test Plan
3/27/24	SRS 6.0	Skyler, Jose	Updated Architecture Design. Added Data Management Strategy

Document Approval

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

Signature	Printed Name	Title	Date
J.L.	Jose Liborio	Software Eng.	2/14/24
S.V.	Skyler Varney	Software Eng.	2/14/24
C.V.	Carlos Venegas	Software Eng.	2/14/24
	Dr. Gus Hanna	Instructor, CS 250	

Table of Contents

1. Introduction

Welcome to the Software Requirement Specification (SRS) document for Rex Movies. This document outlines the requirements for the development of a web-app for Rex movies. Rexmvovies.com will be the starting point for the expansion and development of Rex Movies. This document will address the requirements for such development and improvement of the company and its stakeholders.

This SRS serves as the foundational blueprint for Rex Movies' digital transformation, encapsulating the essential elements necessary for the creation of a progressive web application. By delineating the functional and non-functional requirements, this document sets the stage for the seamless integration of advanced technologies and immersive content offerings both functionally and non-functionally, to pave the way for smoothly blending in the latest tech and captivating content, assisting in a new era of entertainment excellence for Rex Movies and its discerning audience.

1.1 Purpose

The purpose of this document is to create a detailed outline of the functionality of Rex Movies' online ticketing system for users and administrators. This SRS document should serve as a set of instructions and requirements for designing the program to be as efficient and effective as possible for the developer, software, and user.

1.2 Scope

Essentially, the scope includes an E-ticketing software for the Movie Theater, Rex Movies, that screens popular films. It includes recording information of tickets, and seats. Providing and updating information of ticket prices, food and beverages, and deals. Showcasing movies available and seats available per movie screening. The System uses various Modules, such as the Movie Management Module that will help keep track of Movie information, by adding and updating Movies on a day-to-day basis.

1.3 Definitions, Acronyms, and Abbreviations

2FA	Two Factor Authentication.
CRM	Customer Relationship Management.
GDPR	General Data Protection Regulation
MPA	Motion Picture Association

PCI DSS	Payment Card Industry Standard
WCAG	Web Content Accessibility Guideline
DBMS	Database Management System
RDBMS	Relational Database Management System

1.4 References

References:

- Cprime. "How to Create a Clear, High-Quality SRS Document." Cprime Blog, cPrime, [02/07/2024].
- IEEE Computer Society. "IEEE Std 830-1998: IEEE Recommended Practice for Software.
- Perforce. "How to Write a Software Requirements Specification (SRS) Document." Perforce Blog, Perforce Software, Inc., [02/07/2024].
- Requirements Specifications." Sponsored by the Software Engineering Standards Committee. IEEE, 20 October 1998.

1.5 Overview

Rex Movies SRS offers a comprehensive guide for the development of its web application, detailing essential requirements, functionalities, and constraints. The document provides an overview of the product's perspective, user characteristics, and general constraints, emphasizing adaptability for diverse user demographics and ensuring compatibility with older systems. Specific requirements are meticulously outlined, covering external interface requirements, functional features, non-functional attributes, and use cases. These sections offer an explicit understanding of the software's interaction with users, hardware, and communication systems, ensuring clarity and alignment among stakeholders throughout the development process.

In addition to detailing functional aspects like ticket purchasing, movie display, and administrator functions, the SRS also addresses non-functional attributes such as performance, reliability, security, and portability. Use cases provide insights into real-world scenarios of user interactions with the system, offering stakeholders valuable perspectives on user behavior and system functionality. The document also considers design constraints, logical database requirements, and miscellaneous factors, ensuring a holistic approach to software development. Overall, the SRS serves as a foundational blueprint, guiding the development of the Rex Movies web application towards success by providing clear requirements and constraints for stakeholders to follow

The rest of SRS will provide a general description of Rex Movies Web application functionality, including the data and functional requirements of the software product. The Product Functions

and User characteristics will be covered in Section 2. Later on, Section 3 will cover Specific Requirements including functional requirements and External Interface requirements.

2. General Description

The general factors that might affect the new web-app for Rex Movies and its requirements include security considerations, user interface design principles, regulatory compliance, and compatibility with various devices and operating systems. While specific requirements will be detailed in subsequent sections, this section provides an overview of the broader considerations that influence the development process. By understanding these factors, stakeholders can gain insight into the overarching goals and constraints guiding the app's design and implementation

2.1 Product Perspective

This subsection outlines a more specific objective for the future of our product and aligns with our goal as a growing company. The aim is for this product to assist Rex Movies in expanding its reach, attracting more of our current audience, and reaching new demographics. Additionally, we aim to make our services more accessible for our older audience.

Rex Movies is committed to deepening its services by adhering to industry standards while also focusing on innovation to enhance customer satisfaction. As part of this commitment, new functions will be added, with a membership feature being one of them.

2.2 Product Functions

Users:

- Account Creation (user)
- Advance Booking Window (user)
- Assigned vs Open Seating (user)
- Bot Protection (system)
- Browser-Based Platform (user)
- Capacity Handling (system)
- Discount Support (user)
- Discount Ticketing (user)
- Maximum Ticket Purchase (user)
- Nearby Theaters Option (user)
- Online & In-person Access (user)
- Post-Purchase Feedback (user)
- Refund Process (user)
- Regular vs Deluxe Tickets (user)
- Review Scraping (system)

- Subscription Option (user)
- Ticket Purchase Deadline (user)
- Ticket Purchase Limits (user)
- Ticketing Timer (user)
- Web Compatibility (user)

Administrators:

- Administrator Mode (admin)
- Concurrent User Capacity (admin)
- Current System Evaluation (admin)
- Database Integration (admin)
- Database Structure (admin)
- Project Budget (admin)
- Prototype Deadline (admin)
- Ticket Sales Data (admin)
- Scalability Concerns (admin)
- Showtimes per Theater (admin)
- Movie Review Integration (admin)

2.3 User Characteristics

The goal of this is to ensure that eventual users have a convenient and effortless experience throughout the web-app, irrespective of their technological knowledge and/or age. Users are expected to have very specific needs during online purchasing, as well as concerns for their privacy and security. Therefore, Rex Movies must guarantee the user's peace of mind.

Furthermore, the user base may include individuals with disabilities or special needs. This underscores the importance of ensuring accessibility and inclusivity in the product design. This could involve providing alternative methods for inputting information, offering support for assistive technologies, and adhering to accessibility standards to ensure that all users can navigate the platform comfortably and efficiently. Overall, as stated prior, the user's experience and satisfaction come as a priority.

2.4 General Constraints

- Age limit
    - Rex Movies fully complies with the MPA standards, therefore it must be advised to our users that any ticket purchased must be adequate to the user's age. No refund will be given to those who violate that rule.
- Usage of Bots
    - A bot identifier (hCaptcha, BotDetect CAPTCHA, KeyCAPTCHA, TextCAPTCHA, SweetCAPTCHA) will be integrated into the checkout session of the website to avoid the usage of bots.
- Ticket Limit

    - Users must be logged to purchase tickets online and its maximum ticket purchase is 20 tickets every 24 hours.
    - Guest purchases are also accepted, however, there is a 2 ticket limit per IP address, per movie.
- Time Limit
    - Preordering tickets must and can only be purchased within 2 weeks of the movie's release date.
    - Tickets can only be purchased up to 10 minutes after the movie has begun.
- Refunds
    - Users can request online refunds up to 30 minutes prior to the movie start time.
    - Staff will be allowed to provide refunds on-site up to 10 minutes prior to the beginning of the function.
- Payment Information
    - One of our top priorities is the security of our users, therefore we will employ the most up to date system of security to ensure our user's privacy and information. Therefore the following measures will be implemented.
        - Encryption
        - Tokenization
        - PCI DSS Compliance
        - Two-Factor Authentication (2FA)
        - Fraud Detection Systems
        - Regular Security Audits
        - Employee Training
        - Restricted Access
        - Secure Development Practices
        - Vendor Security

2.5 Assumptions and Dependencies

- Accessible for old or outdated systems hardware
    - We understand that many of our users' systems might not be up to date with current technologies. Therefore, Rex Movies will ensure the utmost adaptability for those circumstances.

3. Specific Requirements

3.1 External Interface Requirements

The External Interface Requirements gives insight into how the software interacts with users, external systems, hardware, and communication systems. As well as detail how the system will assist the various interfaces.

3.1.1 User Interfaces

The system will provide an aesthetic and friendly web interface for its customers, which will be accessible in any web browser. It will be in accordance with WCAG (Web Content Accessibility Guidelines) standards and provide accessibility features, such as keyboard accessibility, text alternatives, input assistance, and much more. Thus, making the navigation of Rex Movies easier.

3.1.2 Hardware Interfaces

The Software will be able to integrate with printers to print tickets or barcodes after purchasing tickets. As well as integrate with receipt printers at Movie Theater locations.

3.1.3 Software Interfaces

The software will integrate with a database management system (DBMS) in order to store and retrieve important information regarding tickets, seats, and movies. It will also integrate with Stripe, one of the top payment gateways, which will safely secure credit/debit card payment processes online or in person.

3.1.4 Communications Interfaces

It is important that this software integrates with email services, which will send reservation confirmations, receipts, promotional offers, and two factor authentication (2FA) for users and administrators.

3.2 Functional Requirements

This section describes specific features of the software project.

3.2.1 Ticket purchase

3.2.1.1 The system must allow for ticket purchases.

3.2.1.2 The system will take the movie, number of tickets, seats, discount, and time as inputs.

3.2.1.3 Our payment system will process the valid payment information.

3.2.1.4 The system will display and email the tickets to the user with a unique code(s).

3.2.1.5 The system must limit the maximum number of bulk ticket purchases to 20.

3.2.1.6 The system must restrict purchases before 2 weeks and after 10 minutes before movie start time.

3.2.2 Display Movie Showtimes and Available Tickets

3.2.2.1 The system must display movies, available seat locations, and times.

3.2.2.2 The system should take the selected movie as input.

3.2.2.3 The system would refresh and load information on tickets.

3.2.2.4 The system should output the available seats and times for the requested movie.

3.2.2.5 The system should let the user know if there are no available seats.

3.2.3 Administrator Mode

3.2.3.1 The system must allow for administrators to be added and for administrator-only functions.

3.2.3.2 An employee must have specified administrator roles.

3.2.3.3 The system should give a user administrator access if their account has specified admin access.

3.2.3.4 A system administrator must be able to refund tickets and see the history of customer purchases.

3.2.3.5 If a user tries to access administrator functions, then access will be denied and an error message will be displayed.

3.2.4 Customer Feedback System

3.2.4.1 The Customer feedback system will allow for feedback from customers for in-person experience at the movie theater regarding amenities, movie selection, and satisfaction.

3.2.4.2 The system allows feedback through kiosks in person, and automatic emails sent days later after ticket purchases.

3.2.4.3 The system tracks each customer's feedback and will showcase ratings of the customer’s experience, which will integrate with a customer relationship management (CRM) system.

3.2.4.4 The system will then provide reports and trends through the CRM system, which will allow for areas of improvement.

3.2.4.5 The system will handle any processing or submission errors, and provide error messages when it occurs.

3.2.5 Display movie reviews and critic quotes

3.2.5.1 The system should display audience reviews and critic quotes.

3.2.5.2 The user must select a movie to see its reviews.

3.2.5.3 The system should show reviews in random order and calculate the average review rating.

3.2.5.4 The system should display reviews.

3.2.5.5 If no reviews or quotes are found the system should display to the user that there are no reviews or quotes respectively.

3.2.6 User accounts

3.2.6.1 The system should allow for account creation and signing in.

3.2.6.2 The system should take username and password as input.

3.2.6.3 The system should check the login credentials match in the database.

3.2.6.4 The system should direct the user to the home page when the credentials are accepted.

3.2.6.5 If the inputted credentials are incorrect an error message should be displayed.

3.2.7 Refunds

3.2.7.1 The user should be able to submit refund requests for ticket purchases.

3.2.7.2 The system should take user account number and ticket purchase ID as input.

3.2.7.3 The system will delete the reserved seat from the movie object and transfer the money back through their payment method.

3.2.7.4 The system displays that the review is being processed and will receive a confirmation email when complete.

3.2.7.5 If there is an error during the refund process, the system will communicate to the user that there was an error and the refund was incomplete.

3.2.8 Discounts (Admin)

3.2.8.1 Discounts for students, military, and seniors should be available and customizable.

3.2.8.2 The system will take the movie ticket price, discount, and dates as input.

3.2.8.3 The system should make discounts available during purchases to customers.

3.2.8.4 The system should output the discounts to the user during purchase and should notify the user that discount verification will be checked in person.

3.2.8.5 If no discounts are available the system should notify the user during purchase.

3.2.9 Movie Search

3.2.9.1 The system should allow users to search for movies and showtimes.

3.2.9.2 The system should take movie names as input.

3.2.9.3 the system should pull information about seat availability, times, and reviews from the database.

3.2.9.4 The system should display movie showtimes, available seats, and movie reviews to the user.

3.2.9.5 If the movie doesn't exist the system will display to the users that no search results were found.

3.2.10 Movie Creation (Admin)

3.2.10.1 Administrators should be able to create movie objects that will contain, screening time, seat availability, and movie reviews.

3.2.10.2 The system should take the movie name as input, screening times and dates, available seats, and location as input.

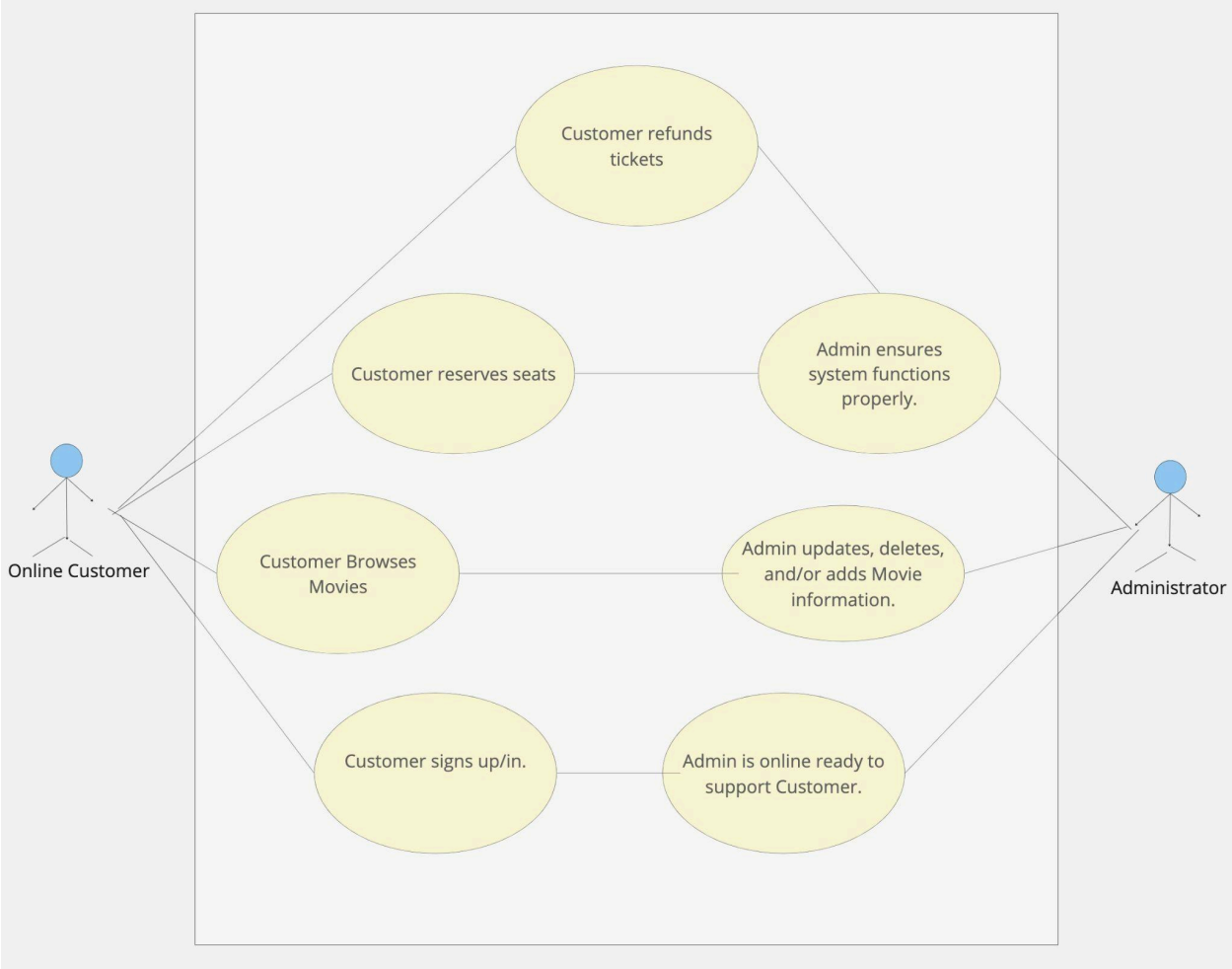
3.2.10.3 The system should create a movie object that contains all attributes stated in the input.

3.2.10.4 The system should display movies to the users during movie search.

3.2.10.5 If the object creation doesn't contain one of the stated inputs, the system should display to the administrator that there was an error during movie creation.

3.3 Use Cases

3.3.1 Use Case Diagram



3.3.2 Use Case #1

Use Case #1: Reserving Seat
Actor: Casual Movie Goer
Brief Description: New customer looked online for nearby movie theaters to watch movies, and they click on Rex Movies.
Stakeholders: Rex Movie Business Owners, Managers, Administrators
Success Guarantee: Customer buys tickets and is left satisfied.
Minimal Guarantee: Customer reaches the seat options screen.
Precondition: Rex Movies website is up and running.
Flow of events:

1. Customer is on the home screen.
2. Customer searches for his desired movie under tab “Movies” or browses through the homepage.
3. Once a movie is chosen, the customer chooses the “Get Tickets” button on the page.
4. Customer selects the best Movie theater location.
5. Customer selects desired movie time.
6. Customer selects the desired seat.

7. Customer selects only one Adult ticket.
8. Customer selects the “Continue” button and is redirected to the checkout to enter credit/debit card information and name.
9. Once done, Customer receives an email confirmation with ticket information.

Extensions:

2a.1 Web failure of any sort.
3a.1 Website is unresponsive.
4a.1 Location services are off.
5a.1 Website is unresponsive.
6a.1 Website is unresponsive.
7a.1 Website is unresponsive.
8a.1 Checkout screen never loads.
9a.1 Email confirmation is not sent.

3.3.3 Use Case #2

Use Case #2: Refunding Ticket
Actor: Recurring Customer
Brief Description: Frequent Customer purchases tickets for a Movie at the wrong time and would like to refund them.
Stakeholders: Rex Movie Business Owners, Managers, Administrators
Success Guarantee: Customer is refunded the full amount.
Minimal Guarantee: Customer is able to submit a refund request.
Precondition: Monthly member has submitted a refund request online and visits a nearby Movie theater location. Administrator is logged into the web-app with valid credentials.
Flow of events:

1. Monthly Member locates the “Contact Support” button on the menu inside the movie theater.
2. Monthly Member shows their refund request number to the help desk.
3. Admin ensures the refund request number is valid.
4. Customer states they would like to change tickets to another time.
5. Admin enforces policy, thus the customer has to buy tickets again.
6. Admin validates Members ID and Credit/Debit card information to refund.
7. Customer is given a refund receipt.

Extensions:

2a.1 Refund request number was never sent.
3a.1 Refund request number doesn’t appear in the data list.
6a.1 Payment gateway isn’t functioning properly.
7a.1 Receipt printer isn’t functioning.

3.3.4 Use Case #3

Use Case #3: Managing Movie Showtimes
Actor: Administrator at Help Desk

Brief Description: Admin updates information on Movie showtimes on web-app for all Rex Movie theaters.
Stakeholders: Rex Movie Business Owners, Managers, Administrators
Success Guarantee: Admin successfully updates showtimes.
Minimal Guarantee: Admin is able to view existing showtimes.
Precondition: Admin is logged into the website with valid credentials.
Flow of events:

1. Admin access Management site.
2. Admin is asked to select a specific Rex Movie theater location.
3. Admin views existing showtimes at chosen movie theater.
4. The admin selects showtime.
5. System displays information about showtime.
6. Admin modifies needed information, such as availability of seats.
7. Admin confirms edits once done.

Extensions:

1a.1. Admin experiences connectivity issues.
3a.1. Admin experiences system errors.
6a.1. Admin adds a new showtime that conflicts with another showtime.

3.4 Classes / Objects

3.4.1 Users

3.4.1.1 Attributes For ‘Account Creation’

- Username
- Email Address
- Password (encrypted)

3.4.1.2 Functions For ‘Account Creation’

- Create Account
- Sign In
- Reset Password

3.4.2 Payment

3.4.2.1 Attributes For ‘Ticket Purchasing’’

- Movie
- Number of Tickets
- Seats
- Time

3.4.2.2 Functions For ‘Ticket Purchasing’’

- Enable Ticket Purchases
- Accept Inputs (Movie, Number of Tickets, Seats, Time)
- Process Transaction
- Display Tickets
- Email Tickets with Unique Code
- Enforce Bulk Ticket Purchase Limit (Maximum of 20)
- Restrict Purchases (Before 2 weeks and after 10 minutes before movie start time)

3.4.3 Tickets

3.4.3.1 Attributes For ‘Ticket Refunding’

- Refund Request Number
- Member ID
- Credit/Debit Card Information

3.4.3.2 Functions For ‘Ticket Refunding’’

- Locate "Contact Support" Button
- Show Refund Request Number
- Validate Refund Request Number
- Enforce Policy (if necessary)
- Validate Member ID and Payment Information
- Issue Refund Receipt

3.4.4 Feedback

3.4.4.1 Attributes For ‘Feedback System’

- Customer Feedback
- Ratings of Customer Experience

3.4.4.2 Functions For ‘Feedback System’

- Accept Feedback for In-Person Experience (Amenities, Movie Selection, Satisfaction)
- Allow Feedback Submission via Kiosks and Automatic Emails
- Track Individual Customer Feedback
- Integrate Feedback Ratings with CRM System
- Provide Reports and Trends for Areas of Improvement
- Handle Processing and Submission Errors, Provide Error Messages

3.4.5 Movie

3.4.5.1 Attributes For ‘Movie Showtimes

- Movie Showtimes
- Availability of Seats
- location

3.4.5.2 Functions For ‘Movie Showtimes’

- Access Management Site
- Select Specific Movie Theater Location
- View Existing Showtimes
- Display Information about Showtime
- Modify Showtime Information (e.g., Availability of Seats)
- Confirm Edits

3.5 Non-Functional Requirements

This section covers the behavioral aspects and the expectations it comes with. It is important to note that non-functional requirements are crucial, such as performance, reliability, availability, security, and more. This system's non-functional requirements should enhance the user's experience with Rex Movies.

3.5.1 Performance

The system must be able to handle 1000 concurrent users without a significant drop in performance.

The speed of the application will depend on the user's hardware, but should be usable from any device.

The web application should load and be available to the user within 10 seconds on supported hardware.

3.5.2 Reliability

The system should remain up and running except for scheduled downtime or maintenance.

The system should have a mean time between failures of at least 99.5%.

In the event of a system failure, the recovery time should be less than 15 minutes.

3.5.3 Availability

The system should have an uptime of at least 99.5%.

Scheduled downtime should be communicated to users in advance.

The application should be accessible from any web browser.

3.5.4 Security

3.5.4.1 Data Storage

User authentication should be stored using secure encryption algorithms.

3.5.4.2 Bot Protection

The system should utilize strategies like ReCaptcha and email verification to minimize or eliminate the use of bots.

The system should regularly be updated to handle new kinds of bot prevention for evolving threats.

3.5.5 Maintainability

The system should be modular to allow for updates to individual components without affecting the entire system.

Provide documentation for system maintenance procedures and troubleshooting.

3.5.6 Portability

The system should be compatible with popular web browsers such as Safari, Chrome, Firefox, etc.

The system should be accessible on multiple operating systems like Linux, Mac OS, and Windows.

3.6 Inverse Requirements

State any *useful* inverse requirements.

3.7 Design Constraints

Specify design constraints imposed by other standards, company policies, hardware limitations, etc. that will impact this software project.

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

List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable to the SRS's requirements.

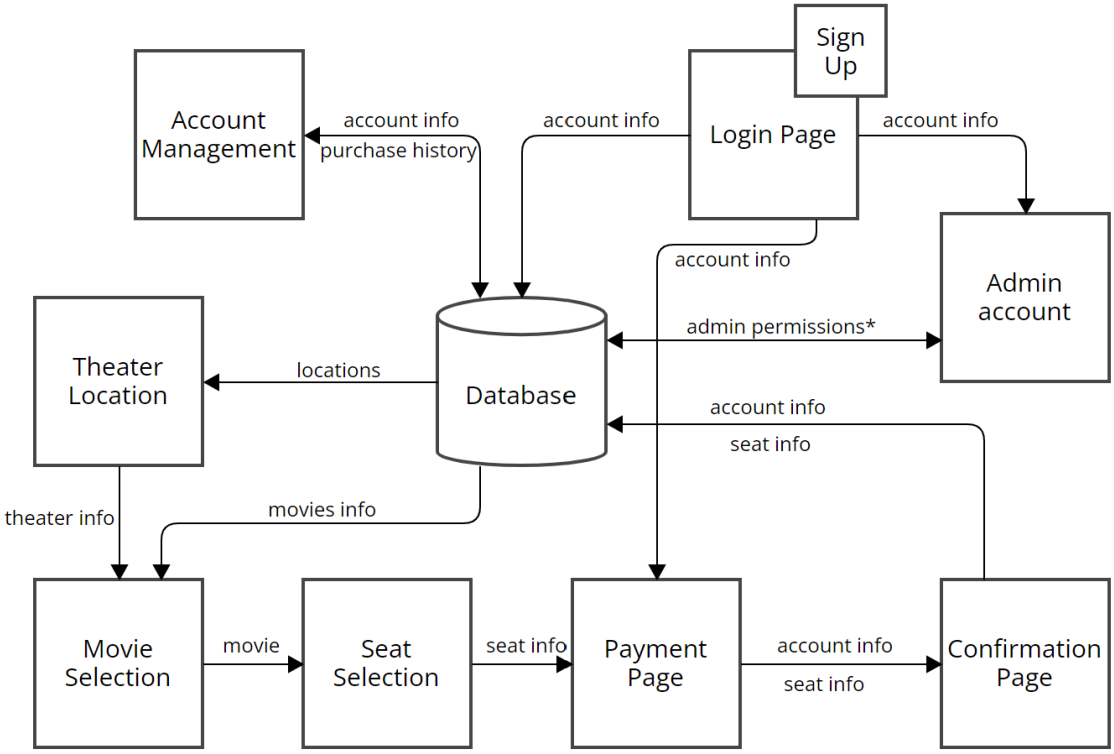
4.1 Sequence Diagrams

4.2 Data Flow Diagrams (DFD)

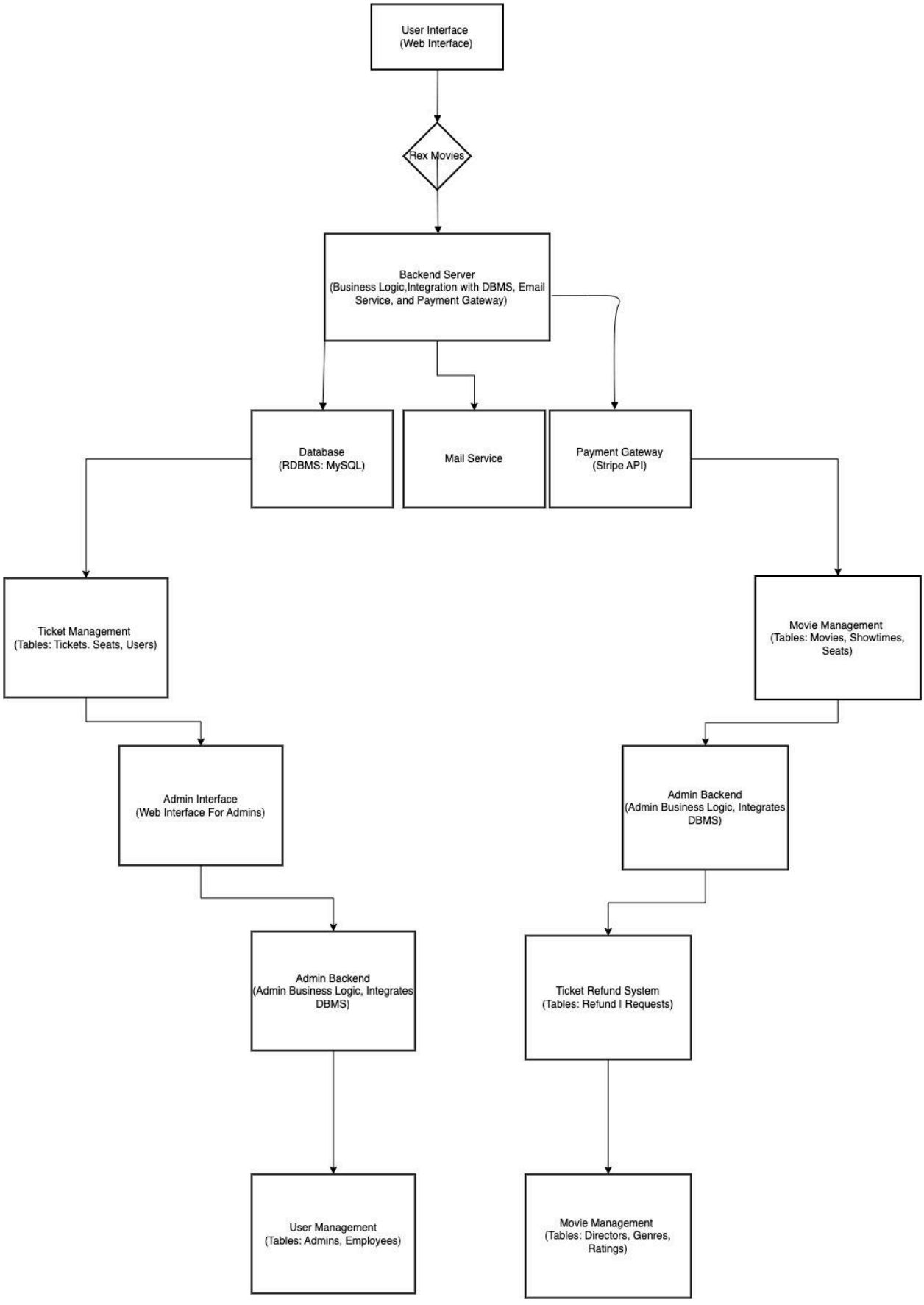
4.3 State-Transition Diagrams (STD)

4.4 SWA Diagram

Our Software Architecture diagram serves as a comprehensive visual representation, elucidating the system's architecture, components, and interactions, thereby facilitating a deeper understanding of its operation and design. At its core, the SWA diagram illustrates the flow of activities between actors and external services. We have meticulously crafted it in such a way that it effectively captures the functionality of this e-ticketing software.



admin permissions*: account info, purchase history, movie info, theater info


User Interface
(Web Interface)
Rex Movies
Backend Server
(Business Logic,Integration with DBMS, Email Service, and Payment Gateway)
Database
(RDBMS: MySQL)
Mail Service
Payment Gateway
(Stripe API)
Ticket Management
(Tables: Tickets. Seats, Users)
Movie Management
(Tables: Movies, Showtimes, Seats)
Admin Interface
(Web Interface For Admins)
Admin Backend
(Admin Business Logic, Integrates DBMS)
Admin Backend
(Admin Business Logic, Integrates DBMS)
Ticket Refund System
(Tables: Refund I Requests)
User Management
(Tables: Admins, Employees)
Movie Management
(Tables: Directors, Genres, Ratings)

4.4.1 SWA Description of Components & Connectors

Front End (Image 1):

- Theater Location: The page will receive the available locations from the database and allow the user to pick a theater location.
- Movie Selection: The theater and movie info will be sent to the movie selection page and let the user select the movie.
- Seat Selection: The seat selection will receive the movie info and display the available seats to the user.
- Payment Page: The payment page will receive the seat and account info and process and let the user confirm the payment.
- Confirmation Page: The confirmation page will display that the seat and account information has been received and the seat is reserved.
- Account Management: The user is able to change password, email, contact info, and is able to see payment history.
- Login/Sign up Page: User is prompted to login or sign up and submits their account info and credentials.
- Admin Account: Admin has permissions to the database, which accesses other components such as account info, purchase history, movie history, and theater info.

Back End (Image 2):

- **User Interface (Web Interface for Customers):** This component represents the front-end interface that customers interact with to browse movies, purchase tickets, and manage their accounts.
- **Backend Server (Business Logic, Integration with DBMS, Email Service, and Payment Gateway):** The backend server hosts the core business logic of the application, handling user requests, processing transactions, integrating with external services, and managing data.
- **Database (RDBMS: MySQL):** The database stores persistent data related to users, tickets, movies, and administrative information.
- **Payment Gateway (Stripe API):** The payment gateway facilitates secure online payment processing for ticket purchases.
- **Email Service:** This component handles the sending of transactional emails such as reservation confirmations, receipts, and promotional offers to users.
- **Admin Interface (Web Interface for Administrators):** This component provides a dedicated web interface for administrators to manage movie listings, view sales data, and perform administrative tasks.
- **Admin Backend (Admin Business Logic, Integration with DBMS):** The admin backend hosts business logic specific to administrative tasks, including managing movie listings, processing refund requests, and generating sales reports.
- **Ticket Management (Tables: Tickets, Seats, Users):** This module within the database manages data related to ticket purchases, seat allocations, and user information.
- **User Management (Tables: Admins, Employees):** This module within the admin database handles administrative user accounts, including administrators and employees.

- **Movie Management (Tables: Directors, Genres, Ratings):** This module within the admin database manages data related to movie listings, including directors, genres, and ratings.

4.4.2 Data Management Strategy

- **User Interface:** We chose to have a web-based interface as it offers accessibility across various devices and platforms, ensuring a broad user reach.
- **Backend Server**: We opted for a server-based architecture to centralize business logic and facilitate seamless integration with various components such as the database, email service, and payment gateway.
- **Database:** We chose MySQL as the relational database management system (RDBMS) due to its robustness, scalability, and support for complex queries. Its relational model suits our data structure well, enabling efficient storage and retrieval of relational data.
- **Email Service:** We integrated an email service to automate communication with users, enhancing user experience and providing essential information regarding their transactions.
- **Payment Gateway:** We chose the Stripe API for its reputation as a reliable and secure payment gateway, ensuring smooth and secure transactions for users while adhering to industry standards for online payments.
- **Admin Interface**: We designed a separate admin interface to streamline administrative workflows, providing administrators with the tools and functionalities they need to manage the platform effectively.
- **Admin Backend**: We implemented a separate backend for administrators to maintain code modularity and separation of concerns, allowing for easier maintenance and scalability of the system.
- **Ticket Management**: We structured ticket management to ensure efficient tracking of reservations, seat availability, and user details, enabling smooth ticketing operations and personalized user experiences.
- **User Management:** We centralized user management for administrators to ensure secure access control and streamlined administrative workflows, facilitating effective management of personnel and permissions.
- **Movie Management:** We structured movie management to organize and categorize movie data, facilitating efficient search and filtering for users and administrators and ensuring comprehensive movie listings.

4.5 UML Diagram

Our Unified Modeling Diagram offers an in-depth structural representation of the system, and serves as a comprehensive visual representation of the system. At the surface level it showcases the classes and objects, and depicts the sequence of interactions between components. For instance, one sequence it depicts is a customer initiating a ticket purchase, which encompasses the interactions with the user interface, backend services, and external databases. In essence, our UML diagram serves as a comprehensive blueprint for the software system.

Functions:
- enableTicketPurchases(): Turns on the ticket purchase process.
- acceptInputs(): Takes in user inputs for ticket purchase.
- processTransaction(): Handles payment transactions.
- displayTickets(): Shows purchased tickets.
- emailTickets(): Sends purchased tickets to users via email.
- enforceBulkPurchaseLimit(): Makes sure bulk purchase limits are followed.
- restrictPurchases(): Puts restrictions on ticket purchases.

**Class: Tickets
Attributes:
- refundRequestNumber: String - Holds the refund request number.
- memberID: String - Stores the member's ID.
- cardInfo: String - Keeps the payment card details.

Functions:
- locateContactSupportButton(): Helps users find the support button.
- showRefundRequestNumber(): Displays the refund request number.
- validateRefundRequestNumber(): Checks the validity of the refund request number.
- enforcePolicy(): Make sure refund policies are followed.

- validateMemberIDAndPayment(): Checks if the member ID and payment info are valid.
- issueRefundReceipt(): Generates a receipt for refunds.

**Class: FeedbackSystem
Attributes:
- customerFeedback: String - Stores customer feedback.
- customerRatings: int - Holds customer ratings.

Functions:
- acceptFeedback(): Collects feedback from customers.
- allowSubmissionViaKiosks(): Lets customers give feedback through kiosks.
- trackIndividualFeedback(): Keeps track of individual feedback.
- integrateWithCRM(): Connects feedback with Customer Relationship Management (CRM) system.
- provideReportsAndTrends(): Generates reports and trends based on feedback.
- handleErrors(): Deals with any errors that may occur.

**Class: MovieShowtimes
Attributes:
- showtimes: List<DateTime> - Holds the list of movie showtimes.
- availableSeats: int - Tracks available seats for each showtime.
- location: String - Stores the theater location.

Functions:
- accessManagementSite(): Gets access to the management site for showtimes.
- selectTheaterLocation(): Allows selection of the theater's location.
- viewExistingShowtimes(): Displays current showtimes.
- displayShowtimeInfo(): Shows details about a specific showtime.
- modifyShowtime(): Allows changes to showtime details.
- confirmEdits(): Confirms any edits made to showtime information.

4.6 Development Plan & Timeline

For Section 4, Skyler Varney is in charge of the SWA Diagram. Carlos Venegas is in charge of the UML DIagram. Jose Liborio is in charge of the descriptions of classes, functions, and attributes. Overall, each member is responsible for ensuring work is complete and correct, ensuring work correlates and allings with software system design. Each is held accountable for work done. Began on February 21, 2024. Held meeting on February 25, 2024. Finished section 4 on February 28, 2024.

4.7 Verification Test Plan

This Test Plan outlines the testing strategy for the e-ticketing software system designed for Rex Movies. The purpose of this document is to ensure that all aspects of the software, including

functionality, performance, security, and usability, are thoroughly tested to meet the specified requirements and deliver a high-quality product to the end-users.

4.7.1 Scope

The testing scope encompasses all components and features of the e-ticketing software system, including but not limited to:

- User registration and authentication
- Ticket booking and purchase
- Seat selection and assignment
- Payment processing
- Integration with movie schedules and showtimes
- Admin functionalities for managing theaters, movies, and user data

4.7.2 Testing Objectives

The main objectives of testing are:

- Validate that the e-ticketing software system meets all specified requirements.
- Identify and rectify any defects or issues in the software.
- Ensure the software is reliable, secure, and performs optimally under various conditions.
- Verify compliance with industry standards and best practices.
- Assess usability and user experience.

4.7.3 Test Cases

The Test Cases will cover various scenarios. Test cases will be categorized based on components.

TS1: Testing Account Creation

- **Description:** Ensure account creation process goes smoothly by allowing users to create accounts without errors.
- **Test Set/Vector:**
    - Valid Account Information
- **Expected Outcome:** The user should be able to create, access, and manage their account successfully.
- **Pass/Fail Criteria:** The test is successful if the account is created and the user is able to sign in and manage it.

TS2: Testing Ticket Purchases

- **Description:** Ensures users are able to successfully purchase a movie ticket.
- **Test Set/Vector:**
    - Valid Payment Method
    - Valid Showtimes
- **Expected Outcome:** Tickets are purchased and the user receives a confirmation email.
- **Pass/Fail Criteria:** The test is successful.

TS3: Testing Refund Request Submission

- **Description:** Verifies refund requests are submitted and processed by the system to give a refund to the customer.
- **Test Set/Vector:**
    - Account Details
    - Valid Payment Method
    - Ticket purchased
- **Expected Outcome:** A refund should be accepted by the system and processed to return money to the payment method.
- **Pass/Fail Criteria:** Refund request will be submitted and processed and a confirmation email will be received.

TS4: Testing Displayed Movie Showtimes

- **Description:** User/Admin is able to view showtimes.
- **Test Set/Vector:**
    - User/Admin choose Theater location
    - Admin has valid credentials
- **Expected Outcome:** User/Admin can search, view, and select movie showtimes.
- **Pass/Fail Criteria:** Showtime is displayed.

TS5: Testing Administrator Mode Access

- **Description:** To ensure administrators can sign in with their account details and have access to administrator only functions.
- **Test Set/Vector:**
    - Valid Account details
- **Expected Outcome:** The administrator logs in and has access administrator mode without issues.
- **Pass/Fail Criteria:** The administrator had access to administrator mode.

TS6: Testing Web Compatibility

- **Description:** User is able to access websites across various browsers, with each functioning correctly.
- **Test Set/Vector:**
    - User has stable connection
- **Expected Outcome:** User successfully manages to browse through the website across different browsers.
- **Pass/Fail Criteria:** User accesses websites across multiple browsers.

TS7: Testing Display Movie Reviews & Critic Quotes

- **Description:** Ensures movie review and critic quotes are displayed correctly to the user under the specified movie.
- **Test Set/Vector:**
    - Movie details
- **Expected Outcome:** Movie reviews and quotes are visible to the customer to read.

- **Pass/Fail Criteria:** Reviews and quotes are displayed properly.

TS8: Testing Guest Ticket Purchase Limit
- **Description:** User attempts to purchase an exceptional amount of tickets.
- **Test Set/Vector:**
    - User has valid payment method
    - User chooses movie showtime
- **Expected Outcome:** System prevents users from purchasing tickets twice from the same IP address.
- **Pass/Fail Criteria:** User makes one payment only.

TS9: Testing Editing Movie Showtimes
- **Description:** Verify that administrators can edit and change movie showtimes
- **Test Set/Vector:**
    - Administrator mode accessed
    - Movie Details
- **Expected Outcome:** The administrator can access and edit the movies' showtime.
- **Pass/Fail Criteria:** the change is updated to the system and displayed to the user.

TS10: Testing Customer Feedback System
- **Description:** Customer leaves feedback of Movie, Service, and/or Food through the in-built feedback system in the website.
- **Test Set/Vector:**
    - Customer is logged in
    - Customer locates feedback system
- **Expected Outcome:** Customer successfully submits their feedback and receives an email notification that their feedback has been recorded.
- **Pass/Fail Criteria:** Customer receives email notification.

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.

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