Movie Ticket Website
“Epic Tickets”

Software Requirements Specification

Version 3

October 23, 2024

Group #1

Jorge Arellano
Efrain Avendano-Gutierrez
Jay Southwick
Noorman Amanuel

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

Fall 2024

Revision History

Date	Description	Author	Comments
9/19/2024	Version 1	Jorge Arellano	First Revision
10/08/2024	Version 2	Jay Southwick	Second Revision
10/10/2024	Version 2	Group #1	Software Design Specification
10/24/2024	Version 3	Group #1	Test Cases

Document Approval

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

Signature	Printed Name	Title	Date
	Dr. Gus Hanna	Instructor, CS 250	
	Pranav Gogawale	Assistant	

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification document is to present a detailed description of the Epic Tickets website application. It outlines the system's purpose, features, user interface (UI), and the functionality it will provide. It also describes the constraints, restrictions, and limitations under which the system will operate, including how it will interact with users and respond to external systems or stimuli. EPic Tickets is a website where users can browse current movie showtimes, choose seats, and buy tickets.

1.2 Scope

The scope of this document defines the key functionalities and constraints of the EPic Tickets website application. This system aims to provide a streamlined and user-friendly platform for purchasing movie tickets, managing user accounts, and ensuring compliance with relevant legal and privacy regulations.

Key Functionalities:

1. **Web Framework and Hosting:**
    a. Built using the Next.js web framework.
    b. Hosted on Google Cloud Platform (GCP) for scalability.
    c. Domain registered via Namecheap.
2. **Core Software Features:**
    a. **Movie Information:** Display high-definition posters, movie titles, genres, runtimes, and synopses.
    b. **Trailers:** Hyperlinks to and embeds official trailers on YouTube.
    c. **Search Functionality:** A search bar for easy navigation.
    d. **Seat Selection:** Visual seating layout that allows users to select and reserve seats for up to 10 minutes.
    e. **Payment Methods:** Support for Visa, Discover, PayPal, Apple Pay, and Google Pay via Stripe.
3. **Account Management:**
    a. Users can create accounts, manage profiles, and save credit cards.
    b. Email confirmations for ticket purchases and invoices via Stripe.
    c. R-rated movie purchase constraints.
    d. Passkeys for Biometric Authentication to enable secure login.
    e. **Administrative Access:** Admins can manage refunds, cancellations, user accounts, and system performance.
4. **Guest Checkout:**
    a. Guest access without account creation is available.

Compliance and Legal Requirements:

- Compliance with California's privacy regulations, including the California Consumer Privacy Act (CCPA), and the EU's General Data Protection Regulation (GDPR).
- Payment processing follows Payment Card Industry Data Security Standard (PCI DSS) requirements via Stripe.
- Accessibility compliance with Web Content Accessibility Guidelines (WCAG) 2.1.

Visual Features:

- Visual seating layout with wheelchair-accessible seating options.

- Enforces maximum seating capacity per theater. If full, users are redirected to other theaters showing the same movie or similar genres.
- Integration with Google Maps for directions to theater locations.

Technology Stack:

- Frontend: JavaScript/Next.js. for dynamic and responsive U
- Backend: Java for scale demand.

Scope Limitations:

- The system is limited to a web-based application; no mobile app is planned, but the website will be mobile-responsive for use on smartphones and tablets.
- Ticket sales are subject to individual theater capacity limits.

1.3 Definitions, Acronyms, and Abbreviations

GDPR: **General Data Protection Regulation**. A comprehensive EU regulation that protects the personal data and privacy of EU citizens and residents.
PCI DSS: **Payment Card Industry Data Security Standard**. A set of security standards designed to ensure that all companies that process credit card information maintain a secure environment.
SRS: **Software Requirements Specification**. A document that outlines the intended features, functionality, and constraints of a software system.
UI: **User Interface**. How a user interacts with a computer or software application, including screens, buttons, flows, and menus.
API: **Application Programming Interface**. A set of rules and protocols for building and interacting with software applications, allowing different systems to communicate.
UX: **User Experience**. The overall experience a user has when interacting with a product or service encompasses usability, design, and functionality.
WCAG: Web Content Accessibility Guidelines.

1.4 References

1. Budgen D. Software Design. Pearson Education Limited, 2003
2. Privacy Act https://www.justice.gov/opcl/privacy-act-1974
3. Privacy Act and General Data Protection Regulations
https://www.consilium.europa.eu/en/policies/data-protection/data-protection-regulation.
4. California Consumer Privacy Act (CCPA), https://oag.ca.gov/privacy/ccpa
5.Payment Card Industry Security Standards Council. *PCI DSS Requirements and Security Assessment Procedures, https://www.pcisecuritystandards.org/standards/
6. Google Cloud Platform Documentation. https://cloud.google.com/docs
7. Figma https://help.figma.com/hc/en-us
8. Stripe API Documentation. https://docs.stripe.com/api
9. JMeter. https://jmeter.apache.org/usermanual/index.html
10. New Relic. https://docs.newrelic.com/
11. OWASP ZAP. https://www.zaproxy.org/docs/
12. JUnit. https://junit.org/junit5/docs/current/user-guide/
13. Jest. https://jestjs.io/
14. Cypress. https://docs.cypress.io/app/get-started/why-cypress

1.5 Overview

The remaining sections of this document outline the scope, features, and constraints of the Epic Tickets website application, detailing its core functionalities, user roles, and compliance with legal and privacy regulations. Section 2 provides a general description of the Epic Tickets system, including its web-based architecture, and hosting on the Google Cloud Platform (GCP). Section 3 covers the functional and data requirements, including ticket purchasing, account management, payment methods, and the UI design (e.g., visual seating layout and search bar). Additionally, this section elaborates on the system's legal compliance with GDPR, PCI DSS, and other relevant standards. Section 4 provides supporting information and references, including definitions, acronyms, and key external documents (e.g., privacy regulations and GCP documentation).

2. General Description

General Description of Factors, Issues, and Considerations for the Development and Operation of the Epic Tickets Website: This section clarifies the environment, technologies, constraints, and external factors that may influence the product's external factors that may influence the product's development and operation. It also outlines the stakeholders and users of the proposed solution.

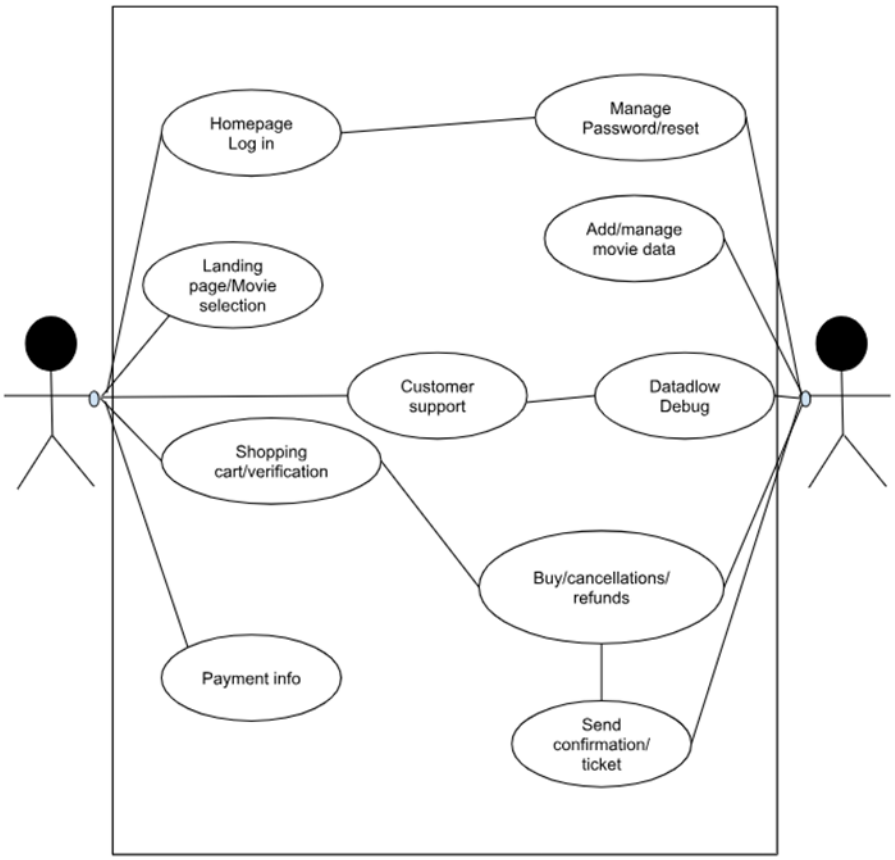


Figure 1

2.1 Product Perspective

The **Product Perspective** section outlines the environment, technologies, constraints, and external factors affecting the development and operation of the Epic Tickets website. It also highlights the common goals shared with competitors and describes how Epic Tickets differentiates itself in key areas such as user experience, technology, and market focus.

2.1.1 General Overview and Market Position

Epic Tickets shares the fundamental goal with competing websites of providing an efficient, user-friendly platform for purchasing movie tickets. In order to scale gracefully, Epic Tickets

provides a streamlined purchasing experience, allowing for biometric authentication. Using our unique platform architecture, we can guarantee 99.9% uptime.

- **Scope and Scale**: The focus of Epic Tickets is a streamlined experience tailored primarily for local theaters, in contrast to competitors in the secondary ticket markets or broader online entertainment services. Related products might offer partnerships with merchandise or other entertainment services.

2.1.2 User Experience

Epic Tickets is designed with a focus on **efficiency and simplicity**, featuring a clean and responsive interface, real-time seat availability, and mobile responsiveness. The platform prioritizes speed, aiming to minimize delays in the purchasing process.

- **Competitor Comparison**: Some competitors offer broader interfaces that include movie news, ticket giveaways, and additional content like online movie rentals and purchases. However, these features may result in a cluttered user interface, longer buffering times, and more frictional process.

2.1.3 Technology Stack

Epic Tickets is designed for **speed and responsiveness**, focusing specifically on movie ticket purchasing. Although the technology stack of related products is not available, it is likely more complex due to their larger scale and the inclusion of additional services. Competitors may experience more **delays and downtime** as a result of their extended service offerings.

2.1.5 Mobile Experience

Epic Tickets offers a **mobile-responsive website** that ensures users can purchase tickets easily on smartphones or tablets, without the need for a dedicated mobile app.

- **Competitor Comparison**: Competitors offer dedicated mobile apps for both iOS and Android, which provide VIP rewards, exclusive content, and engaging user experiences. While this offers more options, it may also introduce additional complexity and slower performance in some cases.

2.1.6 Seat Selection and Theater Interactions

Epic Tickets provides a **visual seat selection tool**, highlighting wheelchair-accessible seats and allowing real-time redirection to other theaters if a movie is fully booked.

- **Competitor Comparison**: Competitors often offer more robust seat selection options, spanning a wider range of theater venues. However, due to their broader integrations with many theaters, they may lack the same flexibility or real-time redirection capabilities that Epic Tickets offers.

2.1.7 Revenue Model

Epic Tickets generates revenue primarily through **service fees on ticket sales** and **payment processing fees**.

- **Competitor Comparison**: Related products generate revenue not only from service fees but also from **advertisements, merchandise sales, online services**, and potential **partnerships with movie studios**.

2.1.8 Market Reach and Partnerships

Epic Tickets targets **local theaters** and **niche groups of moviegoers**. The platform emphasizes close connections with specific markets, prioritizing a more localized experience.

- **Competitor Comparison**: Competitors often have partnerships with major theater chains, providing a wider geographic reach and greater influence across multiple markets.

2.1.9 Content and Extra Features

Epic Tickets primarily focuses on the **purchase of movie tickets**, offering **high-definition movie posters**, **trailers**, and **visual seat selection** for a simplified user experience. It intentionally avoids overwhelming users with additional features, focusing on efficiency.

- **Competitor Comparison**: Competitors often provide extra content such as **critic ratings**, **exclusive interviews**, and **a wider selection of theaters**, making their platforms more content-rich, but potentially more complex.

2.1.10 Compliance and Legal Factors

Epic Tickets is designed to comply with important legal and security frameworks, including **GDPR**, **CCPA**, and **PCI DSS**, ensuring secure payment processing and data protection.

- **Competitor Comparison**: Related products must comply with broader **legal standards** given their partnerships with numerous theaters and handling of ticket resales. They also deal with a wider range of legal concerns, such as refund policies and international data privacy regulations.

2.2 Product Functions

Our product simplifies the process of purchasing movie tickets, allowing users to avoid long lines and skip account sign-ups unless they choose to. It provides real-time information on current and upcoming movies, shows how busy each theater is, and offers seat selection, all at an affordable price. With our user-friendly platform, moviegoers can enjoy a stress-free experience from start to finish.

2.2.1 Register user

Use case: New user creates account

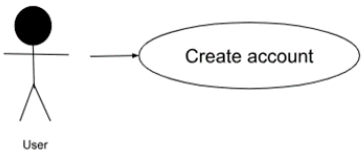


Figure 2

Brief Description

For faster checkout due to saved payment methods the user must register. If the user is not currently registered. The user accesses the "Register" feature from the website's interface.

Initial Step-By-Step Description

- The user will input their information to create an account. (Name, email address, password creation (case sensitive/with strength validation). Biometric authentication (passkey) and unique user verification will be confirmed.
- The user can reset the password.
- The system creates a new account in its database.
- Email confirmation is sent.
- User(s) can access the system**.**
- Registered user selects preferred movie theater, movie, show time, and seat location
- The system saves valid payment method.
- Ticket information is saved to a database of valid tickets.
- Confirms information selected. Booking and QR are sent.
- Email is sent and all data is saved in the user's account.

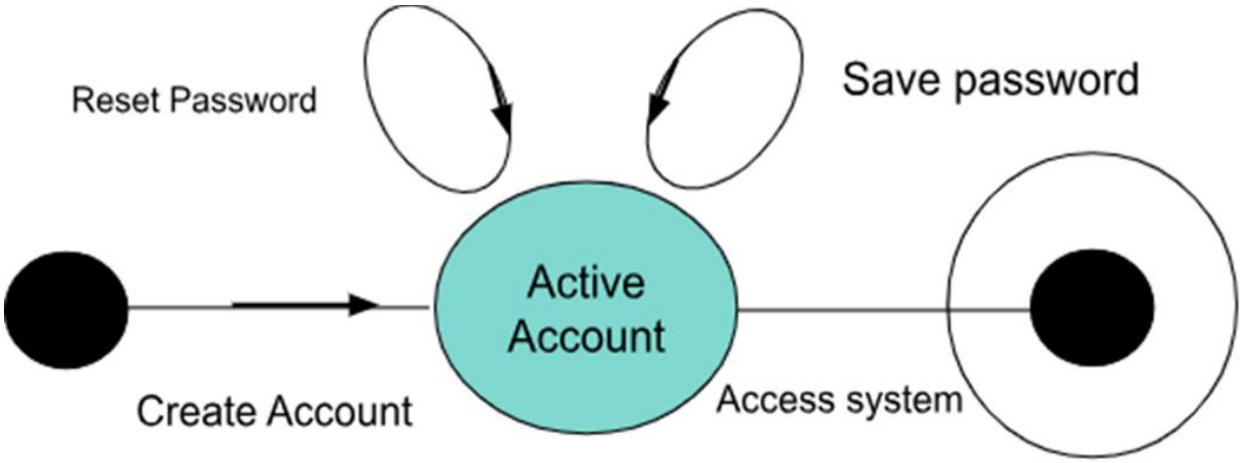


Figure 3

Use case: New user decides not create an account

Brief Description

User is not logged in. User selects the theater, movie and seat(s). Checkout option is available.

Initial Step-By-Step Description

- User selection option to continue as guest.
- User selects preferred movie theater, movie, show time, and seat location.
- Payment is entered and will not be saved.
- User is asked to confirm first and last name and email. Payment information will also be captured if they choose not to use Google Pay or Apple Pay.
- User receives confirmation and ticket.

2.2.2 User

The User after register has the following:

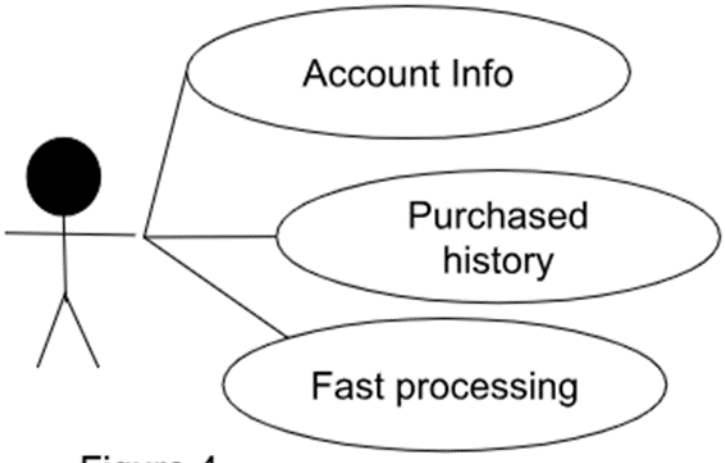


Figure 4

2.3 User Characteristics

This section outlines the primary user roles interacting with the Epic Tickets system, detailing their permissions and interactions with the platform:

2.3.1 Registered User

A registered user has an account on the Epic Tickets platform, allowing them access to features that enhance the user experience through stored preferences, payment information, and purchase history.

Permissions and Interactions:

- **Account Creation and Management:**
  - Create an account with personal details (name, email, password).
  - Update profile information, including adding or removing saved payment methods.
  - Reset forgotten passwords and enable biometric authentication via Passkeys.
- **Ticket Purchasing:**
  - Select a movie, theater, and seat(s) using the visual seating layout.
  - Reserve seats for up to 10 minutes.
  - Seat restriction defaults to 35 per customer.
  - Pay using a stored payment method or enter a new one during checkout.
  - Receive email confirmations for ticket purchases and invoices.
- **Additional Benefits:**
  - Access order history and retrieve past purchases.
  - Get priority access for specific events
  - Use QR codes for easy entry into theaters.

2.3.2 Guest User

A guest user does not need to create an account and can purchase tickets anonymously. The guest checkout process is streamlined to allow quick transactions without requiring registration.

Permissions and Interactions:

- **Ticket Purchasing:**
  - Select a movie, theater, and seat(s) via the visual seating layout.
  - Pay using any supported payment method (Visa, PayPal, Google Pay, etc.).
  - Enter an email address for receipt and booking confirmation.
  - No account creation is required; however, the system will not store payment details or purchase history.

2.3.3 Administrator (Admin)

An admin has access to backend system features necessary for maintaining the website, managing user accounts, and addressing customer service requests.

Permissions and Interactions:

- **System Maintenance:**
  - Manage website performance, ensuring a 99.99% uptime guarantee.
  - Monitor theater capacity limits and enforce redirects to alternative showings when theaters are full.
- **User Management:**
  - Handle user account queries, including password resets, refunds, and cancellations.

    - Approve or deny refund requests and resolve disputes.
    - Sets tickets sold per customer.
    - Manage movie listings, seat availability, and user interactions for compliance with legal standards (e.g., GDPR, PCI DSS).
- **Security and Compliance:**
    - Ensure that all transactions and user interactions adhere to PCI DSS and GDPR standards.
    - Oversee compliance with privacy policies (CCPA, GDPR).

2.4 General Constraints

This section outlines the technical, legal, performance, and business constraints that the Epic Tickets website must adhere to during its development and operation.

2.4.1 Technical Constraints

- **Hosting and Scalability:**
    - The website will be hosted on **Google Cloud Platform (GCP)**, and all systems must conform to GCP's scalability and resource management capabilities.
    - The system is limited to a web-based platform, and no mobile app will be developed. However, the website will be **mobile-responsive** for smartphone and tablet users.
- **Third-Party Rate Limits:**
    - Integration with third-party services, such as payment gateways (Stripe) and Google Maps, will be subject to their respective rate limits however does implement automatic scaling based on detected demand. Changes or updates in third-party APIs may affect performance and availability.
- **Internet Dependency:**
    - The system requires a **reliable internet connection** for users to complete transactions smoothly. Poor connectivity may cause delays or failures in the purchasing process.

2.4.2 Legal and Regulatory Constraints

- **Data Privacy:**
    - The system must comply with **California Privacy Laws (CCPA)** and the **General Data Protection Regulation (GDPR)** to protect users' personal data.
- **Payment Security:**
    - Payment processing must adhere to the **Payment Card Industry Data Security Standard (PCI DSS)** to ensure secure transactions and safeguard users' payment information.
- **Movie Rating Constraints:**

    - Users must meet the required age for purchasing tickets to **R-rated movies**. Verification mechanisms (e.g., biometric, age verification gate, or ID verification) must be in place to enforce this.

2.4.3 Performance and Operational Constraints

- **System Scalability:**
    - The system must be able to handle **sudden traffic spikes** (e.g., during movie premieres), processing transactions within a target time of **two minutes**.
- **Theater Capacity:**
    - The system must enforce **movie theater seating capacities** and redirect users to alternate theaters or showtimes when a theater reaches its maximum capacity.
- **Real-Time Seat Availability:**
    - The seat selection interface must provide **real-time seat availability** and allow users to hold reservations for up to 10 minutes.

2.4.4 Security Constraints

- **Authentication and Authorization:**
    - The system must support secure **biometric authentication** (e.g., Passkeys) for registered users to access their accounts.
    - User data, including payment methods, must be stored securely in accordance with data security best practices.
- **Web Content Accessibility:**
    - The system must comply with **Web Content Accessibility Guidelines (WCAG) 2.1**, ensuring that content is accessible to users with disabilities, including adaptable screen sizes and readable text for screen readers.

2.4.5 Business Constraints

- **Theater Ownership:**
    - The platform will only support movie theaters that are **owned and operated by the company**, limiting partnerships with other theater chains or external vendors.

2.5 Assumptions and Dependencies

Product Functions

1. User Interface and Movie Information:
    - Users can browse and view high-definition movie posters, movie titles, genres, runtime, and a brief synopsis of each movie.
    - Official movie trailers are available as hyperlinks that redirect users to YouTube.
    - A search bar allows users to quickly locate specific movies by title or genre.

- The movie listings page will display available showtimes, ticket pricing, and cinema locations.

2. Ticket Purchasing System:
   - Seating Selection: A dynamic spatial seating layout allows users to view seat availability, select seats, and identify special seating (e.g., wheelchair-accessible).
   - Multiple Payment Methods: The system supports payments through Visa, Discover, PayPal, Apple Pay, Google Pay, and Stripe, ensuring secure transactions compliant with PCI DSS.
   - Real-Time Availability: Tickets will be available for purchase in real-time, with live updates on seating capacity, preventing overselling.
   - Capacity Enforcement: If a theater is fully booked, the system will redirect users to other company theaters nearby that show the same movie.
3. Account Management:
   - User Profiles: Users can create accounts to save personal and payment information, view purchase history, and receive email confirmations/invoices.
   - Credit Card Management: Users can securely save multiple payment options for faster checkouts.
   - Guest Access: Guest users can purchase tickets without creating an account but will have limited features.
   - Passkeys and Biometric Authentication: Users can log in with biometric authentication for faster, secure access to their accounts.
   - R-Rating Constraints: The system restricts ticket purchases for movies with an R-rating, based on the user's age verification.
   - Administrative Functions:
   - Administrators can process refunds, cancel reservations, and manage both user accounts and system performance.
   - The admin dashboard will offer tools for tracking ticket sales, movie performance, and user activity.
   - Admins can also monitor and enforce legal compliance, including privacy regulations like GDPR and California’s no-sale data laws.
5. Email and QR Code Integration:
   - After purchasing, users receive email confirmation and a digital invoice with a QR code via Eventbrite.
   - The QR code can be scanned at the theater for seamless entry.
   - Legal and Security Compliance:
   - The system adheres to the GDPR, California’s privacy regulations, and PCI DSS standards for secure payment transactions.
   - User data will be handled in compliance with these regulations, ensuring secure storage and processing.
   - Additional Features:

- Redirect to Google Maps: The system provides a hyperlink to Google Maps for users to get directions to the selected theater.
- Mobile Responsiveness: Though no mobile app is planned, the website will be fully mobile-responsive to ensure a smooth experience on smartphones and tablets.
- Constraints and Limitations:
- The system is exclusively for the website and will not have a dedicated mobile app, though the website is optimized for mobile use.
- The system will prioritize ensuring real-time performance and compliance with legal and payment regulations but may not support non-cinema events.

3. Specific Requirements

3.1 User Interfaces and System Interfaces

3.1.1 User Interfaces

- **Requirement**: The system shall provide a user-friendly and intuitive interface. It shall be responsive across all devices (desktop and mobile), allowing users to browse and perform tasks seamlessly.
- **Compatibility**: The UI shall be compatible with major browsers (e.g., Chrome, Firefox, Safari).

3.1.2 Hardware Interfaces

- **Requirement**: The system shall be accessible from any device with internet connectivity (desktops, laptops, tablets, and smartphones).
- **QR Scanner**: A hardware interface (QR code scanner) shall be used by the theater staff to validate tickets upon entry.

3.1.3 Software Interfaces

- **Requirement**: The system shall integrate with third-party services as follows:
- **Stripe**: Payment transactions will be facilitated via the Stripe API.
- **YouTube**: Trailers for each movie will be linked via YouTube URLs.

3.1.4 Communications Interfaces

- **Requirement**: The system shall use the HTTPS protocol to ensure secure communication between users and the server, safeguarding all interactions.

3.2 Functional Requirements

3.2.1 Movie Browsing

3.2.1.1 Introduction

- **Description**: Users shall be able to browse available movies, including current listings and upcoming releases.

3.2.1.2 Inputs

- **User Inputs**: Movie titles, categories, filters (genre, length, release dates).

3.2.1.3 Processing

- **Backend**: The system shall query the database for movies matching the user's filters.

3.2.1.4 Outputs

- **Results Display**: A list of matching movies shall be displayed, including: Title, genre, length, movie poster, synopsis, and critic ratings.
  Hyperlinks to official trailers hosted on YouTube.

3.2.1.5 Error Handling

- **No Results**: If no movies match the search filters, display: "Oops… No Movies Found."
- **Database Retrieval Failure**: If data cannot be retrieved, display: "Unable to retrieve data. Please try again later."

3.2.2 Payment Processing

3.2.2.1 Introduction

- **Description**: Users shall be able to purchase tickets for their selected movie and seats.

3.2.2.2 Inputs

- **User Inputs**: Payment information (e.g., credit card or digital wallet).
- **Registered Users**: Option to save payment details for future purchases.

3.2.2.3 Processing

- **Transaction**: The system shall process payments through Stripe.

3.2.2.4 Outputs

- **Success**: Upon a successful transaction, the system shall generate a receipt and a QR code for ticket validation.

3.2.2.5 Error Handling

- **Payment Failure**: In the case of insufficient funds or invalid payment information, display: "Payment failed! Please try again or use a different payment method."

3.2.3 Account Management

3.2.3.1 Introduction

- **Description**: Users shall be able to create and manage their accounts, including saving payment credentials and viewing purchase history.

3.2.3.2 Inputs

- **User Inputs**: Email address, password, and optional biometric authentication.

3.2.3.3 Processing

- **Database**: The system shall store user data securely and allow users to update account details (e.g., email, password, payment methods).

3.2.3.4 Outputs

- **Success**: Users will be able to view their account details, including past purchase history.

3.2.3.5 Error Handling

- **Duplicate Email**: If the provided email is already in use, display: "Email is already linked to an account, please log in or use a different email."

3.2.4 Seat Selection

3.2.4.1 Introduction

- **Description**: Users shall be able to select seats visually for a specific movie and showtime.

3.2.4.2 Inputs

- **User Inputs**: Theater location, movie, showtime, and desired seats.

3.2.4.3 Processing

- **Seat Reservation**: The system shall display a real-time seating layout, highlighting available, reserved, accessible, and selected seats. Once selected, seats will be reserved for up to 10 minutes.

3.2.4.4 Outputs

- **Visual Layout**: Display the user's seat selections and update in real-time.
- **Reservation Timer**: Warn the user with the message "2 minutes left until your reservation expires!" when the timer reaches 2 minutes. If the timer expires, display: "Your reservation time has expired! Please reload the page" and redirect the user to the movie page.

3.2.4.5 Error Handling

- **Seating Data Unavailable**: If seating data cannot be retrieved, display: "Unable to load, please try again later."
- **Seat Selection Error**: If an error occurs with seat selection, display: "Oh no! It seems to be an error with your seat selection, please select new seats."

3.2.5 Guest Checkout

The **Guest Checkout** feature allows users to purchase movie tickets without needing to create an account. This section details the input, processing, outputs, error handling, constraints, and security considerations associated with this process.

3.2.5.1 Inputs

The system shall accept the following inputs for guest checkout:

- **Theater Selection**: Guests select the theater where they want to watch the movie.
- **Movie Selection**: Guests choose the movie they want to watch.
- **Showtime Selection**: Guests select the specific showtime.
- **Seat Selection**: Guests choose their preferred seats from the visual seating chart.
- **Payment Information**: Guests input payment details, including credit/debit card information, or choose from options like PayPal and Google Pay.
- **Email Address**: An email is required for sending the confirmation and QR code for the ticket(s).

3.2.5.2 Processing

The system shall perform the following processing steps:

- **Input Validation**: All inputs, including email and payment information, are validated.
- **Payment Gateway**: The system shall process the payment through Stripe using the same gateway as for registered users.

- **Seat Reservation**: The selected seats will be temporarily held for the guest user for **10 minutes** to allow sufficient time for the payment to be completed.
- **No Data Retention**: Guest checkout sessions will not save payment details for future transactions.

3.2.5.3 Outputs

Upon successful transaction, the system shall provide the following outputs:

- **Confirmation Email**: A receipt and QR code for theater entry will be sent to the provided email address.
- **Confirmation Page**: The system shall display a confirmation page summarizing the transaction details, including the QR code for ticket entry.

3.2.5.4 Error Handling

The system shall handle errors in the guest checkout process as follows:

- If **payment validation fails** (due to invalid payment information or insufficient funds), the system shall display:
  "Payment failed! Please try again or use a different payment method."
- If the **email address** is invalid, the system shall prompt the user to provide a valid email before proceeding.
- If the guest fails to **complete the purchase within the 10-minute seat reservation window**, the system shall release the seats and display:
  "Your reservation time has expired! Please reload the page."

3.2.5.5 Constraints

- **No Purchase History**: Guest users will not have access to a purchase history through the website. All information regarding the transaction will be available only in the confirmation email.
- **No Saved Payment Information**: Payment details will not be stored for future use in guest checkout, requiring the user to re-enter the information for each transaction.
- **Limited Features**: Certain features, such as biometric authentication, saved seat preferences, and early access to ticket bookings, will only be available to registered users.

3.2.5.6 Security and Compliance

The system shall enforce security and legal compliance measures for guest checkout:

- **Data Protection**: The system shall ensure that guest users' email addresses and payment information are handled securely.
- **Compliance**: The guest checkout process shall comply with **GDPR**, **CCPA**, and **PCI DSS** regulations, ensuring that sensitive user information, such as credit card details, is not stored after the transaction.
- **Transaction Security**: Encryption protocols will be used to protect guest data during the transaction process, ensuring a high level of security without requiring the user to register an account.

3.2.6 Ticket Cancellation

The **Ticket Cancellation** feature provides users the ability to cancel their movie tickets prior to the showtime. This section describes how the cancellation process works, the timeframe, refund policies, and how the system handles errors related to cancellations.

3.2.6.1 Ticket Cancellation Process

Users can cancel purchased tickets via two methods:

- **Via Confirmation Email**:
  The confirmation email sent after ticket purchase shall contain a link labeled **"Cancel Tickets"**. Clicking this link will direct the user to the cancellation page.
- **Via Website**:
  Guests can also access the cancellation feature directly through the website by navigating to the **"Manage My Booking"** section. They must enter their **confirmation number** and **email address** to retrieve their booking and proceed with the cancellation.

3.2.6.2 Cancellation Timeframe

- **Cancellation Deadline**:
  The system shall allow ticket cancellations up to **2 hours before the scheduled movie showtime**.
- **Late Cancellation**:
  If users attempt to cancel after this window, the system will display a notification:
  "Cancellations are no longer allowed for this showtime."

3.2.6.3 Cancellation Confirmation

Once a cancellation is successfully processed, the system shall send a **cancellation confirmation email** that includes the following details:

- **Cancellation ID**: A unique identifier for the cancellation.
- **Cancellation Summary**: Details of the canceled tickets, including movie title, showtime, and theater.
- **Refund Information**: The amount refunded (if eligible), any applicable cancellation fees, and the expected time for the refund to be processed (typically **5-7 business days**).

3.2.6.4 Refund Process

Refunds will be processed according to the following policies:

- **Refund Eligibility**:
  Users eligible for a refund will be informed of the amount and whether any non-refundable fees apply (e.g., service or booking fees).
- **Refund to Original Payment Method**:
  Refunds shall be processed back to the original payment method used during the transaction, typically within **5-7 business days**.
- **Refund Policy Display**:
  Clear refund policies, including any non-refundable fees or special circumstances (e.g., promotional offers or discounts), will be displayed prominently on the cancellation page.

3.2.6.5 Error Handling

The system shall handle errors related to ticket cancellation as follows:

- **Cancellation Window Passed**:
  If users attempt to cancel outside the allowed timeframe, the system shall display the error message:
  "Cancellations are no longer allowed for this showtime."
- **Technical Issues**:
  In the event of a technical issue that prevents the cancellation from being processed, users will be advised to contact **customer support** through a link or contact details provided on the error page.

3.2.6.6 Cancellation Policy Display

The **cancellation policy** shall be clearly outlined for users:

- The policy, including cancellation timing restrictions, any **non-refundable items** (such as booking fees), and special conditions, will be displayed on both the **checkout page** and the **confirmation email**.
- This ensures that users are aware of the terms before they proceed with their purchase.

3.2.7 Notification System

The **Notification System** ensures that users receive timely and relevant information related to their transactions, account, and movie showtimes. Notifications help guide users through the purchasing process, confirm important updates, and remind them of upcoming events.

3.2.7.1 Types of Notifications

1. Transaction Notifications

- **Purchase Confirmation**:
  Upon successful ticket purchase, the system shall send a confirmation email to the user's provided email address. This email will contain:
  - A **receipt** of the transaction.
  - A **QR code** for theater entry.
- **Failed Transaction Notification**:
  In the event of a failed transaction, the system will send a notification detailing:
  - The reason for the failure (e.g., insufficient funds or payment method issues).
  - Instructions on how to retry the transaction or contact customer support for further assistance.
- **Refund Confirmation**:
  If a refund is issued due to ticket cancellation or any other reason, the system will notify the user with:
  Details of the refund, including the refunded amount and expected processing time.
- **Ticket Change Notification**:
  Should there be any changes to the purchased tickets, such as a seat reassignment or schedule modification, the system shall send a notification containing:

Updated ticket details.

Instructions or next steps for the user, if necessary.

2. Account Notifications

- **Account Creation Confirmation**:

  When a user successfully creates an account, the system shall send an email containing:

  A **verification link** to confirm the email address used during registration.

- **Password Reset Notification**:

  If a user requests a password reset, the system will send a notification with:

  A secure link allowing the user to **reset their password**.

- **Profile Update Confirmation**:

  When a user makes changes to their account information (e.g., email or phone number), the system will send a notification confirming:

  The **successful update** of the account details.

3. Showtime Reminders

- **24-Hour Reminder**:

  The system shall send a reminder notification **24 hours** before the scheduled movie showtime. The email will include:

  Showtime details (movie title, time, and theater location).

  A reminder to bring the **QR code** for entry.

- **2-Hour Reminder**:

  For last-minute preparedness, the system may send an additional reminder **2 hours** before the showtime. This notification will contain similar information to the 24-hour reminder and help ensure the user is ready for the movie.

This notification system is designed to improve user engagement and ensure that important updates related to transactions, accounts, and events are clearly communicated.

3.2.7.2 Delivery Channels

To ensure that users receive timely and relevant updates, the notification system will support multiple delivery channels, allowing for flexibility and reliability in communication.

1. Email Notifications

- **Transaction-Related Emails**:

  All users, including guests, shall receive emails related to transactions, such as:

  Booking confirmations.

  Cancellations.

  Refund confirmations.

- **Delivery Assurance**:

  The system shall attempt to deliver the email to the address provided by the user. In cases where delivery fails (e.g., due to an invalid email address or temporary server issues), the system will retry delivery and log the failure.

2. SMS Notifications

- **Opt-In for SMS**:
  Users who opt into SMS notifications will receive critical updates like:
  Showtime reminders.
  Transaction confirmations.
  Cancellations.
- **Essential Information Only**:
  SMS notifications shall contain only essential information, keeping the content concise to avoid overwhelming the user. Example content could include reminders to bring the QR code for entry or a simple booking confirmation.

3. Web/In-App Notifications

- **For Registered Users**:
  Registered users logged into the platform will receive notifications within the website's interface. These notifications will be accessible through a dedicated **Notification Center**, where users can:
  View the history of their notifications.
  Check unread or missed notifications.
- **Types of In-App Notifications**:
  Booking confirmations.
  Payment or refund confirmations.
  Seat or schedule change alerts.
  Showtime reminders.

3.2.7.3 Notification Management

1. User Preferences

- **Opt-In/Opt-Out Control**:
  Users will have the option to manage their notification preferences through account settings. This includes:
  Opting in or out of **promotional notifications**.
  Adjusting preferences for **transaction-related notifications** (e.g., choosing between email or SMS).
- **Customization of Notifications**:
  Users shall be able to specify their notification preferences on an **email preferences page**, where they can:
  Select how and when they receive certain types of notifications, particularly for marketing content.
  Adjust delivery methods for transactional notifications.

2. Error Handling

- **Logging and Retry**:
  If a notification fails to send due to an invalid email or phone number, the system shall:
  Log the failure.
  Attempt to resend via the user's **next available contact method** (e.g., if an email fails, it might send an in-app notification or SMS, if enabled).
- **Admin Alerts for Repeated Failures**:
  If delivery fails repeatedly (e.g., due to a persistently invalid email or phone number), the system shall generate an alert to notify the admin, who will investigate and resolve the issue.

This approach ensures that users are kept informed, with flexibility in how they receive notifications, and helps mitigate issues when notifications fail to be sent.

3.2.7.4 GDPR and Privacy Compliance

The notification system must comply with GDPR and any relevant privacy regulations

3.3 Use Cases

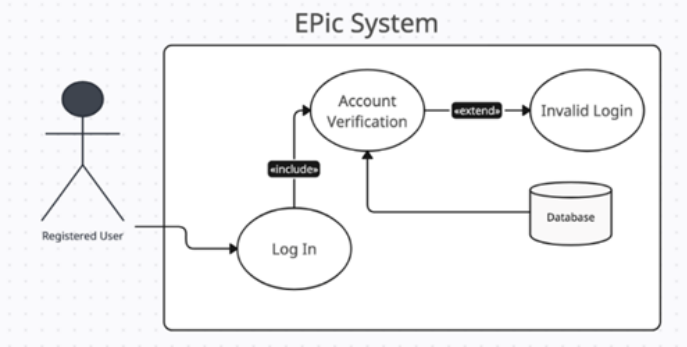
3.3.1 Registered User Login

Name of Use Case: Registered User Login
Actor(s): Registered User
Flow of events:

- The user clicks the Login tab
- The user enters their login credentials (email and password).
- The system checks the database to match the stored account
- If credentials are valid, the account verification is successful and logs in the user.
- If credentials are invalid, the system will prompt an error message.

Entry conditions:

- The user shall have an account.
- Users shall log in utilizing the log-in page from the website.



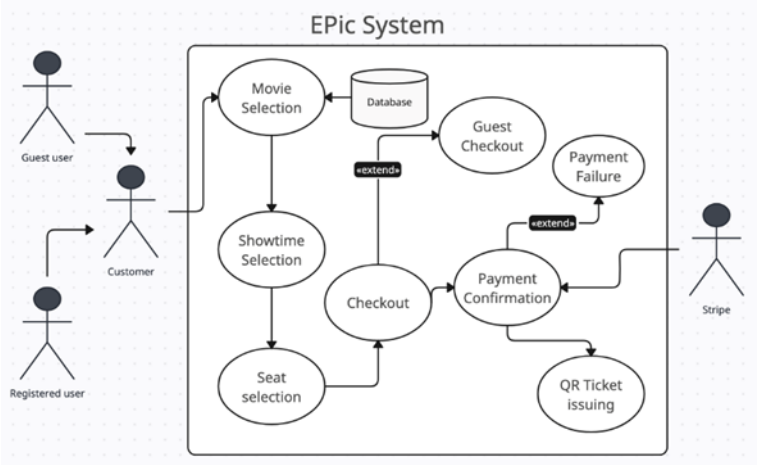
3.3.2 User buying tickets

Name of Use Case: User buying tickets
Actor(s): Customer, Registered user, guest user, Stripe System
Flow of events:

- The customer, whether a registered user or not, selects a movie and showtime listed.
- The customer selects the seats they want to reserve.
- The system reserves the selected seats for 10 minutes during the payment process.
- A registered user shall proceed using their saved payment credentials. A guest user shall enter a contact email and payment information.
- The system sends payment information to the Stripe System which processes the transaction.
- If the translation is a success the system will output a QR ticket and transaction information to the customer's email. If the translation fails, an error message will prompt the customer to use a different payment method.

Entry conditions:

- The user shall select a movie, showtime, and desired seats.
- The user must have a valid payment method.



3.3.3 Ticket Cancellation

Name of use case: Ticket Cancellation
Actor(s): Customer, Stripe System, Administrator
Flow of events:

- The customer browses to the “Returns and Cancellations” tab in the website.
- If the customer has an account, they shall login and select the ticket they want to return.
- If the customer is a guest user, then they may return their tickets by entering the email address that they use for the purchase alongside their receipt number.
- The Stripe system shall issue a refund to the same payment method that the customer used.

- A refund confirmation shall be sent to the customer, and their QR ticket shall be now invalid.
- If an error occurs during the process, then the webpage's administrator may manually issue a refund and override the 2-hour limit if necessary.
- Entry conditions: The user shall not cancel tickets two hours before showtime.
- The user must have purchased a ticket

3.4 Classes / Objects

3.4.1 User

The User object represents individuals interacting with the system. This can be a registered user or a guest user.

Attributes:

- userID: Unique identifier for the user (nullable for guests).
- email: User's email address.
- password: Encrypted password (registered users only).
- name: User's full name.
- isGuest: Boolean indicating whether the user is a guest or a registered user.
- phone: user's phone number
- preferences: User notification and communication preferences (email, SMS, etc.).

Methods:

- register(): Registers a new user account.
- login(): Authenticates a user.
- logout(): Logs the user out of the system.
- updatePreferences(): Updates notification settings.

3.4.1.2 Movie

The Movie object represents movies available for booking on the platform.

Attributes:

- movieID: Unique identifier for the movie.
- title: Title of the movie.
- description: Movie synopsis.
- duration: Duration in minutes.
- genre: Genre of the movie.
- rating: Age rating (e.g., PG, R).
- showtimes[]: List of available showtimes for the movie.
- posterURL: URL of the movie poster.
- theaters: List of theaters available

Methods:

- getShowtimes(): Retrieves available showtimes for a movie.
- getMovieInfo(): Provides detailed information about the movie.

3.4.1.3 Ticket

The Ticket object represents a movie ticket booked through the system.

Attributes:

- ticketID: Unique identifier for the ticket.
- movie: Reference to the Movie object.
- showtime: Showtime for which the ticket is booked.
- seatNumber: Seat assigned to the ticket.
- price: Price of the ticket.
- isPaid: Boolean indicating whether the ticket has been paid for.
- qrCode: Generated QR code for entry.
- transactionID: Associated transaction ID for payment.

Methods:

- generateQRCode(): Generates a QR code for ticket validation.
- cancelTicket(): Cancels the ticket and processes refunds based on policy.
- getTicketDetails(): Retrieves detailed information of the ticket.

3.4.1.4 Transaction

The Transaction object represents the payment and booking details for one or more tickets.

Attributes:

- transactionID: Unique identifier for the transaction.
- user: Reference to the User object.
- tickets[]: List of Ticket objects associated with the transaction.
- totalAmount: Total amount paid for the transaction.
- paymentMethod: Payment method used (credit card, PayPal, etc.).
- isRefunded: Boolean indicating whether the transaction has been refunded.

Methods:

- processPayment(): Processes payment for the transaction.
- refundTransaction(): Processes a refund for the transaction.
- getTransactionSummary(): Provides a summary of the transaction.

3.4.1.5 Notification

The Notification object represents system notifications, which are used to alert users about their bookings, transactions, or promotional offers.

Attributes:

- notificationID: Unique identifier for the notification.

- recipient: Reference to the User object.
- type: Type of notification (e.g., booking confirmation, promotional offer).
- message: The content of the notification.
- timestamp: Date and time the notification was sent.

Methods:

- sendEmail(): Sends an email notification to the user.
- sendSMS(): Sends an SMS notification to the user.
- getNotificationHistory(): Retrieves the user's notification history.

3.4.1.6 Theater

The Theater object represents a specific movie theater.

Attributes:

- theaterID: Unique identifier for the theater.
- name: Name of theater location.
- location: Address of movie theater.
- screens[]: List of screens in a theater.

Methods:

- getTheaterInfo(): Provides information regarding the theater.
- getMovies(): Provides the available movies in the theater.

3.4.1.7 Screen

The screen object represents specific screening rooms within a theater location.

Attributes:

- screenID: Unique identifier for the screening room.
- theaterID: Reference to the Theater object.
- size: Total number of seats available.
- screenType: Type of screen used in room (e.g. IMAX, Standard).

Methods:

- getScreenDetails(): Fetches information about current screen.**3.4.1.8 Seat**

The Seat object represents the seats in a screening room.

Attributes:

- seatID: Unique identifier for the seat.
- screenID: Reference to a screen object.
- row: row location of the seat.
- number: seat number in reference to its row.
- seatType: Type of seat. (e.g. VIP, Accessible).
- isReserved: Boolean, indicates a seat that has been bought. False by default.

Methods:

- getSeatInfo(): Provides information regarding the seat.
- reserveSeat(): Reserves a seat
- freeSeat(): Opens seats that were previously reserved.

Attributes Breakdown for Guests vs. Members

Guests:
userID: null
isGuest: True
name: Required
email: Required
password: null
preferences: null

Registered Members:
Members:
userID: Required
isGuest: False
name: Required
email: Required
password: Required

3.4.3 Functions

This section outlines the core functions that the Epic Tickets website application must provide to fulfill its purpose and meet user needs effectively.

3.4.3.1 User Registration and Account Management

- **Functionality**: Users can create and manage accounts to store personal information, payment methods, and ticket purchase history.
- **Details**:
  Users must provide a valid email address and create a secure password.
  Users can update personal details, manage saved credit cards, and opt in or out of email notifications.
  Account recovery options, including password resets, will be available for user convenience.

3.4.3.2 Movie Browsing and Information Retrieval

- **Functionality**: Users can browse and search for movies available for ticket purchase.
- **Details**:
  Display high-definition movie posters, titles, genres, runtimes, and synopses to provide comprehensive movie information.
  Provide hyperlinks to official movie trailers hosted on YouTube for users to view trailers directly.
  A search bar will facilitate quick access to specific movie information.

3.4.3.3 Seating Selection and Reservation

- **Functionality**: Users can visually select seats within the theater layout during the ticket purchasing process.
- **Details**:

  The seating layout will clearly indicate available, reserved, and wheelchair-accessible seats for better user experience.

  Users can reserve selected seats for 10 minutes to complete their purchase, ensuring they have time to finalize payment.

  After the reservation time expires, the seats will be released back into availability for other users.

3.4.3.4 Payment Processing

- **Functionality**: Users can securely complete their purchases using various payment methods.
- **Details**:

  Support for major credit cards (Visa, Discover) and digital wallets (PayPal, Apple Pay, Google Pay) will be provided.

  Payments will be processed through Stripe, ensuring compliance with PCI DSS standards for payment security.

  Users will receive email confirmations and invoices following successful purchases for their records.

3.4.3.5 Ticket Confirmation and Verification

- **Functionality**: Users receive confirmation of their purchases, enabling ticket verification at the theater.
- **Details**:

  Confirmation codes will be sent via email, which users must present at the theater for entry.

  The system will allow users to view and manage their ticket purchases through their accounts, providing easy access to their purchase history.

3.4.3.6 Administrative Functions

- **Functionality**: Administrators can manage user accounts and oversee system operations.
- **Details**:

  Administrators will have the authority to process refunds and cancellations, ensuring customer satisfaction.

  They will monitor ticket sales, system performance, and user feedback to maintain operational effectiveness.

3.4.3.7 Compliance and Security Features

- **Functionality**: The system ensures compliance with applicable legal and privacy regulations.
- **Details**:

Compliance with GDPR, CCPA, and WCAG 2.1 will be upheld to protect user data and ensure accessibility.
Implementation of biometric authentication will provide a secure and streamlined login process for users.
Regular updates and audits will be conducted to maintain data security and uphold user privacy.

3.4.3.8 User Experience Enhancements

- **Functionality**: The platform aims to provide a seamless and efficient user experience.
- **Details**:
  The website will be fully responsive to ensure accessibility across various devices, including smartphones and tablets.
  Quick loading times and intuitive navigation will facilitate the purchase process, minimizing user frustration.
  Integration of real-time seat availability and notifications for changes will keep users informed and engaged throughout the purchasing journey.

3.5 Non-Functional Requirements

3.5.1 Performance

- **Concurrent Users**:
  The system shall support up to **10,000 concurrent users** without significantly affecting performance.
- **Response Times**:
  Interaction with the site (e.g., browsing movies and locations) shall have a maximum response time of **2 seconds**.
  Seat selection shall be displayed within **3 seconds**.
  Payment confirmations shall generate a confirmation receipt and a unique QR code within **4 seconds**.
  During peak load times, the system shall allow up to **10 seconds** to process any transaction.

3.5.2 Reliability

- **Uptime Target**:
  The system will maintain an uptime target of **99.99%**, allowing for no more than a calculated **10-minute window of unplanned downtime** per week.
- **Maintenance Exclusions**:
  Scheduled maintenance windows are not included in the uptime calculation. Planned maintenance will only take place when theaters are closed.

3.5.3 Availability

- **Public Access**:
  The system shall be available to the public **24/7**, with zero noticeable downtime for maintenance.
- **Live Maintenance**:
  The system shall support live maintenance techniques, such as **rolling updates**.
- **Limited Downtime**:
  If downtime is necessary, it shall be limited to no more than **one hour** during off-peak hours to minimize negative user experience.
- **Real-Time Updates**:
  Movie listings shall be updated in real-time and reflected within **5 seconds**.

3.5.4 Security

- **Compliance Standards**:
  The system shall comply with **PCI DSS** (Payment Card Industry Data Security Standard) to ensure the encryption of sensitive data.
  - **QR Code Security**:
    The system shall generate secure and unique **QR codes** for each ticket purchase, which shall be validated by theater staff upon entry. User passwords are encrypted using 256-bit AES encryption.

3.5.5 Maintainability

- **Scheduled Maintenance**:
  Maintenance windows shall be scheduled during off-peak hours to minimize user impact.
- **Zero-Downtime Maintenance**:
  Given the uptime target of **99.99%**, the system shall implement zero-downtime maintenance practices wherever feasible.

3.5.6 Portability

- **Browser Compatibility**:
  The system must be compatible with all major browsers, including **Safari**, **Edge**, and **Chrome**.
- **Responsive Design**:
  The web application shall be fully responsive to mobile devices, ensuring an optimal user experience without the need for a dedicated app.

These non-functional requirements ensure that the Epic Tickets system is robust, user-friendly, and secure, providing an efficient and reliable platform for users

3.6 Inverse Requirements

The following inverse requirements are designed to ensure the Epic Tickets system operates securely and respects user preferences, enhancing overall user experience and compliance with relevant standards.

No Transaction Without Confirmation

- **Explicit Confirmation**: The system must not complete a ticket purchase without the user explicitly confirming the transaction (e.g., by clicking "Confirm"). This ensures that users are fully aware of their purchases before completion.

No Automatic Refunds Without User Consent

- **User Notification**: Refunds must not be processed automatically without notifying the user and obtaining their explicit consent. This provides transparency and gives users control over their financial transactions.

No Unsolicited Promotional Emails

- **Opt-In Requirement**: The system must not send promotional emails or notifications to users who have not opted into such communication. This respects user preferences and complies with privacy regulations.

No Changes to Booked Tickets Without Notification

- **User Notification**: The system must not alter the details of a booked ticket (e.g., changes to seat assignments) without sending a notification to the user with the updated details. This ensures users are informed of any changes that affect their bookings.

No Permanent Storage of Payment Information

- **Temporary Storage**: Payment details must not be stored permanently unless the user opts for it (e.g., allowing the option to save a card for future use). Compliance with PCI DSS must be ensured to protect sensitive information.

No Display of Sensitive Information in Notifications

- **Data Protection**: Notifications must not display sensitive personal or financial information, such as full payment card numbers, passwords, or other Personally Identifiable Information (PII). This enhances user privacy and security.

No Guest Checkout Without Explicit Consent

- **Limited Functionality Awareness**: Users must not proceed with a guest checkout without being informed of the limited functionality (e.g., inability to save tickets or view purchase history). This ensures transparency regarding user capabilities.

No Execution of Transactions During System Downtime

- **Transaction Halt**: The system must not execute any transactions when critical services (e.g., payment gateways) are unavailable or experiencing downtime. Appropriate user messages must be displayed to explain the situation.

No Hidden Fees or Charges

- **Clear Communication**: The system must not add hidden fees or additional charges without clearly informing the user during the checkout process. This builds trust and fosters user satisfaction.

No Overriding User Preferences for Notifications

- **Respect for Preferences**: User-selected preferences for email or SMS notifications must not be overridden by the system (e.g., sending promotional messages even after opting out). This maintains user trust and ensures compliance with privacy laws.

3.7 Design Constraints

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

Legal Compliance

- The system must follow local laws for content, age ratings, and licensing agreements with theaters and movie studios.

Third-Party Integration

- The system will need to work smoothly with external services like movie databases (e.g., **IMDb**) and payment platforms (e.g., **Stripe**, **PayPal**) while respecting their limitations.

User Account Policy

- Since signing up for an account is optional, the system must support **guest checkout** with minimal data collection. There are terms and conditions all users will be required to agree to before purchase.

Performance

- The software needs to handle high traffic during popular movie releases, ensuring quick response times for seat selection and payments.

Hardware Limitations

- The system should perform well on a wide range of devices or slow internet connections, keeping the site lightweight and fast.

Payments

- All transactions will be in **USD** only, so the software will not need to handle multiple currencies or exchange rates. since its only for theaters in the U.S.

Accessibility

- The software must comply with **accessibility standards** to ensure users with disabilities can easily interact with it (e.g., screen readers, high contrast modes).

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.

Data Formats

- Supports structured data (e.g., movie details, showtimes) and **JSON** for complex data (e.g., seating maps).
- All transactions and prices are stored in **USD**.

Storage Capabilities

- Must scale for large data volumes, especially during peak times.
- Supports real-time updates for seating availability and showtimes.
- Handles multi-theater data efficiently.

Data Retention

- Stores user purchase history and transaction records based on business/legal requirements.
- Archives old movie/showtime data after movie runs.
- Regular data archiving and deletion policies.

Data Integrity

- Ensures atomic transactions for ticket purchases and seat reservations.
- Maintains data consistency for real-time updates.
- Validates input to prevent errors like overbooking.

Security and Privacy

- Sensitive data encrypted at rest and in transit.
- Role-based access control to manage permissions.

Backup and Recovery

- Automated backups and a disaster recovery plan for system restoration.

3.9 Other Requirements

Catchall section for any additional requirements.

Notifications

- Option for users to receive alerts for movie releases and promotions

Offline Access

- Limited features like viewing movie details should be available offline.

Scalability

- Must scale easily to add new theaters or regions.

3.10 Test Plan

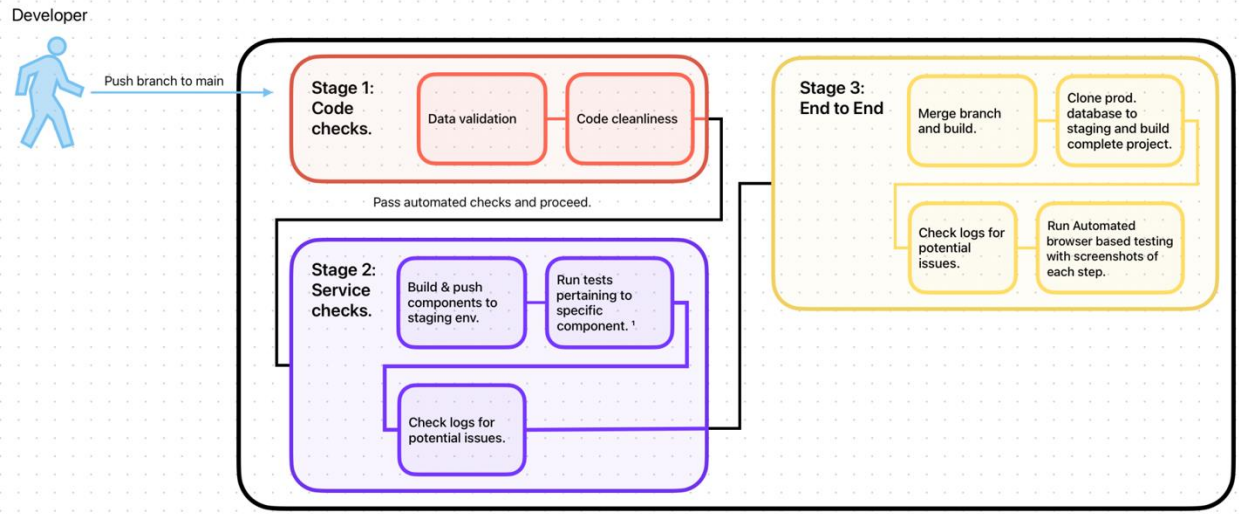
3.10.1 Test Overview

Potential errors cover a wide range of edge cases. While most errors will revolve around erroneous user input, some may involve issues with service communication. To preemptively solve these issues, we use several techniques to test our implementations.

For each build, we run a suite of automated tests before allowing the code to be pushed to our main branch. These tests include code cleanliness, networking testing, expected data validation, and speed differences. These tests let us catch bugs early, meaning we don't have to wait for a version increment to test our smaller changes.

From there, we move to larger scope testing. This includes staged testing of modified components. For example, the service that is responsible for handling ticket validation is updated and staged to verify compatibility. These tests help us isolate issues that may impact specific services.

Finally, our overall systems test relies on a few technologies. Before deploying to production, a branch must first pass an end-to-end test on our staging environment. This environment contains a mirror of the production database allowing us to conduct potentially breaking tests without impacting actual user experience.



3.10.2 Testing Objectives

- **Ensure that all system components meet all necessary requirements for proper integration and functionality.**
- **Confirm security and system compliances.**

Verify system performance, alongside with error handling that works for edge cases.3.10.3 Scope of Tests

The scope of the test plan encompasses all critical components, functionalities, integrations, and performance metrics of the EpicTickets system, ensuring compliance with defined requirements and legal standards. The following areas are covered:

1. **Functional Testing:**
    - Verification of all user interactions, including ticket selection, payment processing, and seat reservation.
    - Testing different user roles (guest users, registered users, admins) to ensure appropriate permissions and access levels.
    - Validation of the checkout flow, including guest checkout and account creation.
    - Handling of promotional codes and discounts.
    - Error handling for invalid user inputs, such as invalid credit card numbers, expired tickets, or session timeouts.
2. **Integration Testing:**
    - Testing the integration between the front-end and back-end services.
    - Ensuring proper communication with third-party APIs, such as payment gateways, theater databases, and email services.
    - Verifying that seat availability syncs correctly with the theater's systems.
    - Testing the response of the system to failed third-party services or network disruptions.
3. **Security Testing:**
    - Assessing the compliance of the system with GDPR and PCI DSS standards.
    - Validation of data encryption during sensitive transactions (e.g., payment processing).
    - Testing user authentication and authorization mechanisms, ensuring secure account management.
    - Penetration testing to identify potential vulnerabilities in the system.
4. **Performance Testing:**
    - Load testing to verify that the system handles peak user traffic, such as during movie releases or ticket promotions.
    - Measuring system response times and database query efficiency under typical and extreme load conditions.
    - Stress testing to determine the system's breaking point and recovery time.
5. **End-to-End Testing:**
    - Ensuring that the user journey from landing on the homepage to receiving a ticket confirmation email functions without errors.
    - Verifying integration between microservices and external systems to validate the full workflow.
6. **User Interface (UI) Testing:**

- Verifying the visual correctness of the ticketing interface across different devices and screen sizes (desktop, mobile, tablets).
- Ensuring accessibility compliance (WCAG 2.1) for visually impaired users, with appropriate screen reader support and keyboard navigation.
- Cross-browser testing to ensure the application behaves consistently across major browsers (Chrome, Safari, Firefox, Edge).

7. **Regression Testing:**
   - Running a suite of regression tests on each release to ensure that newly introduced code does not break existing functionality.
   - Continuous testing of critical workflows such as user login, ticket purchase, and seat selection to detect regressions early.
8. **Database Testing:**
   - Validation of data consistency across user accounts, ticket purchases, and payment records.
   - Ensuring backup and recovery processes function properly without loss of data.
   - Testing for proper handling of database transactions, especially when dealing with concurrent users purchasing tickets for the same event.
9. **Usability Testing:**
   - Gathering feedback from user testing sessions to ensure that the platform is intuitive, especially during tasks such as browsing movie listings, selecting seats, and making payments.
   - Testing for navigational clarity and ease of use across various features.
10. **Compliance Testing:**
    - Ensure compliance to PCI DSS, for data and payment security.
    - Utilizes automated testing security tools to ensure proper encryption of sensitive data, and its proper transfer and storage.

3.10.4 Test Types

3.10.4.1 Unit Testing: Test the functionality of individual components. Automated tests shall be tan for components such as user registration, payments, and seat selections. These tests shall verify and validate input fields, and proper data storage usage.

3.10.4.2 Functional Testing: Test interactions between the system's components.

3.10.4.3 System Testing: Test the functionality of the overall system. The methods for test cases of this nature shall follow a complete use case such as user registration, payments, logins.

3.10.5 Testing tools

Performance Testing Tools: JMeter, New Relic.
Security Testing Tools: OWASP ZAP.
Automated Testing Tools: JUnit, Jest, Cypress.

Test Cases									
TestCaseId	**Component**	**Priority**	**Description/Test Summary**	**Pre-requisites**	**Test Steps**	**Expected Result**	**Actual Result**	**Status**	**Test Executed By**
UserTC001	User Registration	P0	Verify that a new user can successfully register with valid credentials.	User is on the registration page	1. Enter valid data for all required fields (name, email, password, etc.). 2. Click on "Register".	The user is successfully registered, notification sent, provided option home/profile	User successfully registered notification sent, option for home/profile provided	Pass	JorgeA
UserTC002	User Authentication	P0	Validate that a registered user can log in with valid credentials.	The user is registered in the system	1. Go to the login page. 2. Enter the correct email & password. 3. Click on "Login".	The user successfully logs in & direct to main page.	Validation successful redirection accomplished	Pass	JorgeA
SearchTC003	Movie Search	P1	Ensure that the user can search for a movie by its title & see the relevant details.	Movies are available in the database.	1. Enter a movie title in the search bar. 2. Click "Search".	The search results display the correct movie & its relevant details (e.g., title, rating, schedule).	Correct movie is returned	Pass	JorgeA
SelectionTC004	Seat Selection	P1	Verify that a user can select available seats from the visual seating layout & reserve for the aloted time.	The user is on the seat selection page.	1. Click on available seats in the seating layout. 2. Confirm seat selection	The selected seats are highlighted & confirmed for purchase.	Available seats reserved for the aloted time.	Pass	JorgeA
CheckoutTC005	Guest Checkout	P0	Ensure that a guest user can purchase tickets without registering an account.	The movie is available for booking, & the user is a guest.	1. Select a movie & available seat. 2. Proceed to checkout. 3. Complete payment using valid payment details.	The purchase is successful, & the guest user receives a confirmation & QR code.	User successfully purchases selection & receives with QR code.	Pass	JorgeA
PaymentTC006	Payment Gateway	P0	Validate that payment fails when incorrect or invalid card information is used.	The user has a invalid payment method at the payment stage.	1. Enter invalid credit card details. 2. Verify information 3. Click "Pay".	The system shows a payment failure message & requests valid card information.	Payment fails & prompts user to enter valid card information	Pass	JorgeA
LogoutTC007	User Session Management	P2	Verify that the user can log out from their account successfully.	The user is logged in.	1. Click the "Logout" button in the account dashboard.	The user is logged out & redirected to the homepage.	User successfully logs out.	Pass	JorgeA
NotificationsTC008	Email Notification System for	P0	Ensure that an email confirmation is sent after a	The user or guest has completed the purchase &	1. Complete the ticket purchase.	The user receives an email with the QR ticket	User promptly receives notification with QR ticket.	Pass	JorgeA

	purchased confirmation		successful ticket purchase with QR ticket.	entered their email address.	2. Check the registered email for the confirmation.	details & confirmation.			
PasswordTC009	User Authentication Password recovery	P1	Validate that a registered user can recover their password via email or text.	The user is on the login page & has forgotten their password.	1. Click "Forgot Password". 2. Enter the registered email. 3. Follow the steps in the email to reset the password.	The user is able to reset their password & log in with the new credentials.	User successfully reset password & able to log in.	Pass	JorgeA
TimeoutTC010	User Session Management Timeout hadling	P2	Verify that the system properly handles session timeout & redirects the user to the login page after inactivity.	The user is logged in & has been inactive for a more than 30 minutes	1. Log in to the account. 2. Remain inactive for 30 minutes or longer. 3.Attempt to perform an action (e.g., search for a movie or select a seat).	The system redirects the user to the login page with a session timeout message	User is logged out after no activity longer than 30 minutes.	Pass	JorgeA
TimeTC011	Web Performance (Front-end and back-end)	P2	Test to ensure website pages open in less than 2 seconds under normal traffic conditions.	Functional environment, browser compatibility, network connection & stability.	1. Select compaitble browser & open website URL. 2. Time total time for complete load time for each pase, link operations, & purchase process. 3. Repeat for different browsers.	Website loads fully (all visible elements) in less than 2 seconds across desktop and mobile platforms.	All pages, links, & purchasing pages loaded in less than 2 seconds.	Pass	JorgeA

4. Analysis Models

This section presents the structural and dynamic models of the Epic Tickets system. It aims to provide an understanding of how different system components interact to fulfill the functional and non-functional requirements outlined. The models described below are based on the use cases and specific requirements of the system.

The primary goal of the Analysis Model is to:

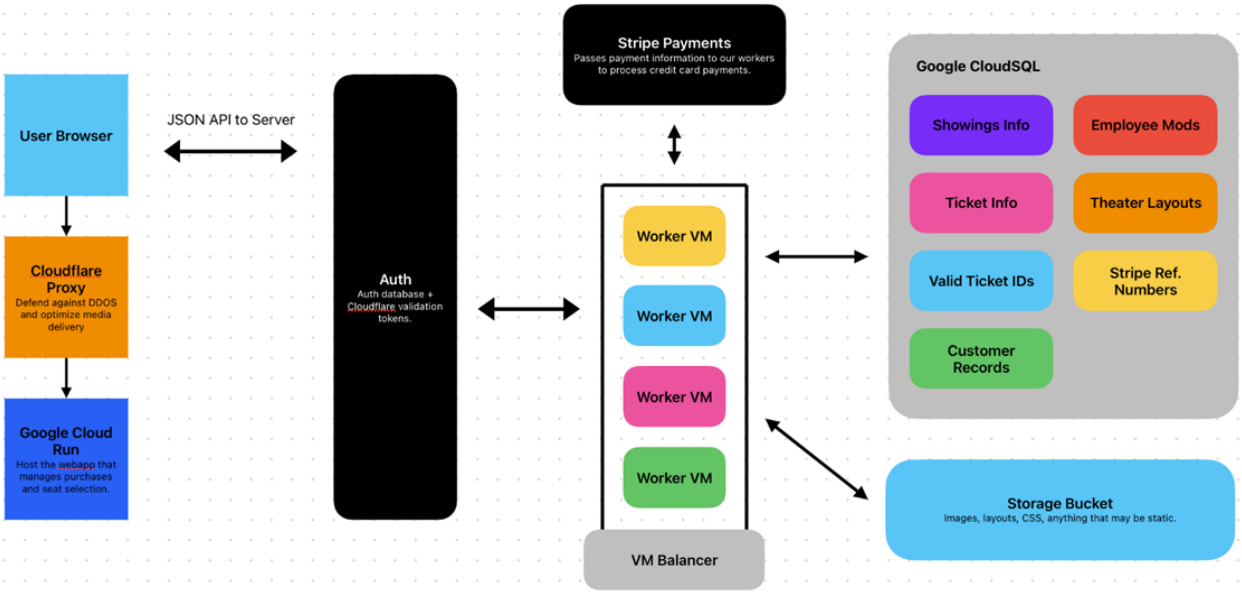
- Define the system's structure through various diagrams (e.g., class diagrams, entity-relationship diagrams).

- Illustrate the interactions between different components (e.g., use case diagrams, sequence diagrams).

- Provide a clear and concise representation of how the system will meet the functional and non-functional requirements.

4.1 Sequence Diagrams

Sequence diagrams illustrate how system components interact over time to perform specific functions. These dynamic models focus the system flows, such as:



1. **User Interaction:** A user's web browser sends a JSON API request to the server.

2. **Cloudflare Proxy:** The request passes through the Cloudflare proxy, which provides DDoS protection and optimizes media delivery.

3. **Authentication:** The server authenticates the user using Cloudflare validation tokens.

4. **API Processing:** The server processes the JSON API request and retrieves necessary data from the Google CloudSQL databases (Showings Info, Employee Mods, Ticket Info, Theater Layouts, Valid Ticket IDs, Stripe Ref. Numbers, Customer Records).

5. **Worker VM Processing:** The server distributes the request to a worker VM within the Google Cloud Run environment.

6. **Payment Processing (if applicable):** If the request involves a payment, the worker VM passes payment information to Stripe for processing.

7. **Data Retrieval and Processing:** The worker VM retrieves additional data from the Google CloudSQL databases as needed and performs any necessary calculations or logic.

8. **Response Generation:** The worker VM generates a response to the user's request, which includes the requested information or the result of the action (e.g., successful payment, ticket purchase).

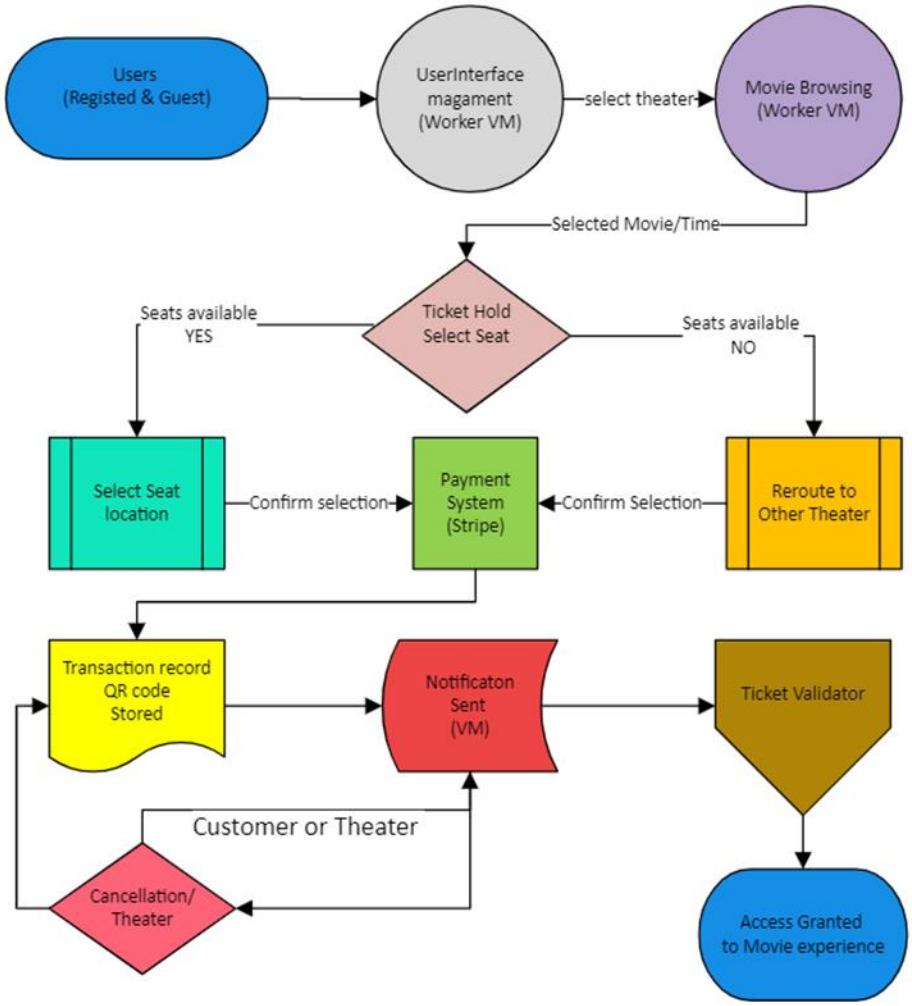
9. **Response Transmission:** The response is sent back to the user's web browser through the Cloudflare proxy.

Key Components and Their Roles:

- **User Browser:** Initiates the interaction by sending a request.
- **Cloudflare Proxy:** Provides network security and optimization.
- **Server:** Handles the request, authenticates the user, and distributes it to worker VMs.
- **Worker VMs:** Perform the actual processing of the request, including data retrieval, calculations, and response generation.
- **Google CloudSQL:** Stores data for the application, such as showings, tickets, customer information, and payment details.
- **Stripe:** Processes payment information.
- **Google Cloud Run:** Manages the worker VMs and scales the application based on load.
- **Storage Bucket:** Stores static assets like images, layouts, and CSS.

This sequence provides a general overview of how the system functions.**4.2 Data Flow Diagrams (DFD)**

The system's interaction with the user and external components during the movie ticket purchasing process is outlined step by step. Here's a quick introduction to this sequence:



Data Flow Sequence:

1. **User Initialization:**
    - **Start:** The process begins with a user, either a registered user or a guest, accessing the movie ticket purchasing system.
2. **Movie Browsing:**

- **User Interface Management:** The user interface (UI) manages the browsing experience, possibly involving a worker VM for dynamic content or interactions.
- **Theater Selection:** The user selects the desired theater.
- **Movie Browsing:** The system presents available movies and showtimes.

3. **Seat Selection:**
    - **Movie/Time Selection:** The user chooses a specific movie and showtime.
    - **Seat Availability Check:** The system checks seat availability for the selected movie and time.
    - **Seat Selection:** If seats are available, the user selects their desired seats.
4. **Ticket Hold:**
    - **Seat Hold:** The selected seats are temporarily held for a certain duration (e.g., 10 minutes) to allow the user to complete the purchase.
5. **Payment:**
    - **Payment System:** The user proceeds to the payment page and enters their payment information.
    - **Stripe Integration:** The payment system integrates with Stripe to process the payment.
6. **Confirmation:**
    - **Confirmation:** Upon successful payment, the system confirms the transaction and generates a transaction record and QR code.
    - **Notification:** A notification is sent to the user, possibly through a worker VM or an external notification service.
7. **Ticket Validation:**
    - **Ticket Validator:** The user presents the QR code at the theater entrance for validation.
    - **Access Granted:** If the QR code is valid, the user is granted access to the movie.
8. **Cancellation/Theater Management:**
    - **Cancellation:** The system allows users to cancel their tickets, potentially with certain restrictions or fees.

- **Theater Management:** The theater can manage ticket sales, cancellations, and other administrative tasks.

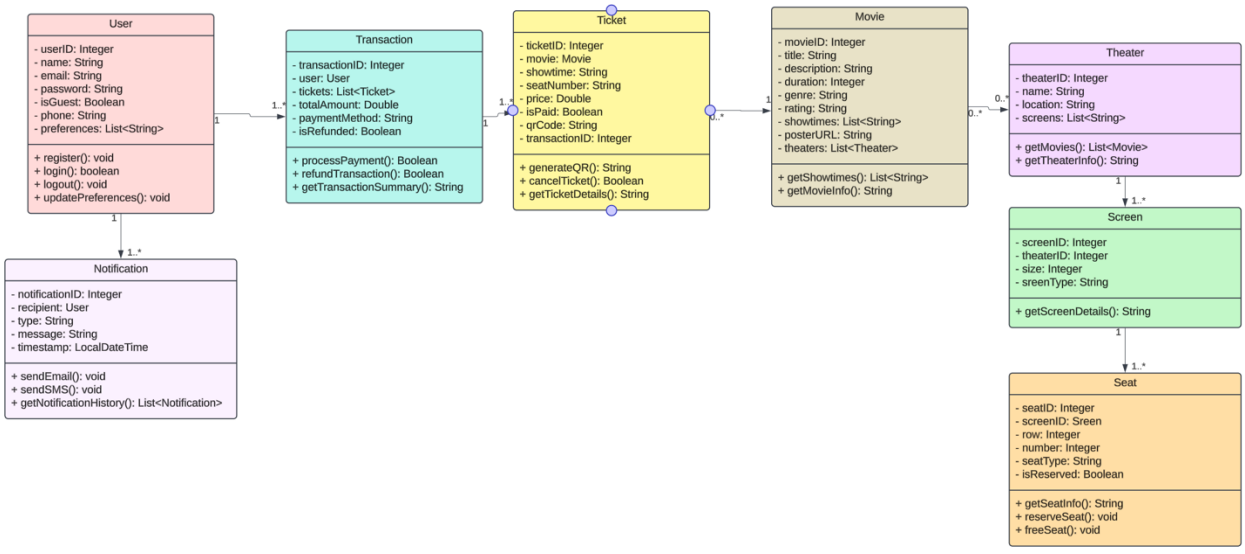
Key Points:

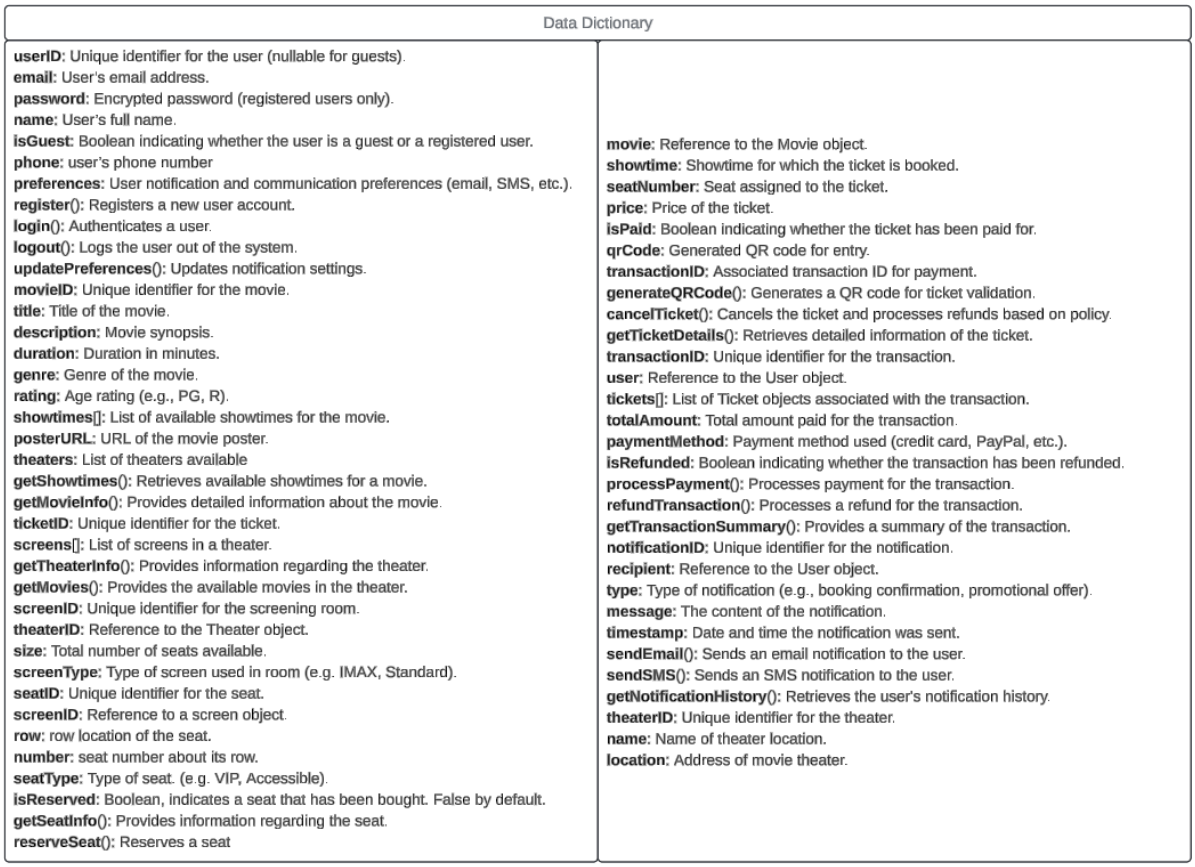
- **User Interface:** The UI plays a crucial role in guiding the user through the process and providing a seamless experience.
- **Worker VMs:** The use of worker VMs is a scalable architecture that can handle multiple requests concurrently.
- **External Services:** Stripe is used for payment processing, while a notification service (not explicitly shown) is likely used for sending notifications.
- **Ticket Validation:** The QR code system provides a convenient and secure way to validate tickets at the theater.

This breakdown provides a general understanding of the data flow in the movie ticket purchasing Epic Ticketing System.

4.3 State-Transition Diagrams (STD)

This section outlines the various states a user can move through while navigating the Epic Tickets movie ticket purchasing system, detailing the transitions between them based on user actions.





1. State: Home Page

- **Transition**: User navigates to the website.
    - **Next State**: Browsing Page
- **Transition**: User clicks "Log In".
    - **Next State**: Login Page
- **Transition**: User clicks "Sign Up".
    - **Next State**: Registration Page

2. State: Browsing Page

- **Transition**: User filters movies by genre, title, or other criteria.
    - **Next State**: Filtered Movie Results
- **Transition**: User clicks on a movie title.
    - **Next State**: Movie Details Page
- **Transition**: User initiates seat selection.
    - **Next State**: Seat Selection Page

- **Transition**: User adds movies to their watchlist.
    - **Next State**: Updated Browsing Page

3. State: Login Page

- **Transition**: User enters valid credentials.
    - **Next State**: Account Home
- **Transition**: User enters invalid credentials.
    - **Next State**: Login Error Page (Message: "Incorrect login credentials. Please try again.")
- **Transition**: User opts to reset the password.
    - **Next State**: Password Reset Page

4. State: Movie Details Page

- **Transition**: User clicks "Watch Trailer".
    - **Next State**: YouTube Interface (opens in new window)
- **Transition**: User clicks "Buy Tickets".
    - **Next State**: Showtime Selection Page
- **Transition**: User clicks "Back".
    - **Next State**: Browsing Page

5. State: Showtime Selection Page

- **Transition**: User selects showtime and theater.
    - **Next State**: Seat Selection Page
- **Transition**: User clicks "Cancel".
    - **Next State**: Movie Details Page

6. State: Seat Selection Page

- **Transition**: User selects desired seats.
    - **Next State**: Payment Page (If guest checkout is selected, no need for account login.)
- **Transition**: User cancels the selection.
    - **Next State**: Showtime Selection Page
- **Transition**: Seats are reserved for more than 10 minutes.

    - **Next State**: Reservation Timeout Page (Message: "Your reservation has expired! Please select seats again.")

7. State: Payment Page

- **Transition**: User enters valid payment details and submits.
    - **Next State**: Confirmation Page (Success message, QR code, and receipt)
- **Transition**: User enters invalid payment details.
    - **Next State**: Payment Error Page (Message: "Payment failed! Please try again or use a different payment method.")
- **Transition**: User cancels the payment process.
    - **Next State**: Seat Selection Page

8. State: Account Home

- **Transition**: User accesses purchase history.
    - **Next State**: Purchase History Page
- **Transition**: User updates account details.
    - **Next State**: Account Details Update Page
- **Transition**: User clicks "Logout".
    - **Next State**: Home Page

9. State: Confirmation Page

- **Transition**: User clicks "View Tickets".
    - **Next State**: Ticket QR Code Page
- **Transition**: User clicks "Cancel Tickets".
    - **Next State**: Ticket Cancellation Page
- **Transition**: User clicks "Back to Home".
    - **Next State**: Home Page

10. State: Ticket Cancellation Page

- **Transition**: User cancels tickets before the 2-hour window.
    - **Next State**: Cancellation Confirmation Page (Success message, refund details)
- **Transition**: User cancels tickets after the 2-hour window.
    - **Next State**: Cancellation Error Page (Message: "Cancellations are no longer allowed for this showtime.")

11. State: Notification System (Background State)

- **Transition**: Successful transaction.
    - **Next State**: Send Confirmation Email & QR Code
- **Transition**: Failed transaction.
    - **Next State**: Send Payment Failure Notification
- **Transition**: Showtime reminder (24-hour).
    - **Next State**: Send 24-hour Reminder Email/SMS
- **Transition**: Showtime reminder (2-hour).
    - **Next State**: Send 2-hour Reminder Email/SMS

12. State: Error Handling

- **Transition**: Database retrieval failure during movie browsing.
    - **Next State**: Error Message (Display "Unable to retrieve data. Please try again later.")
- **Transition**: Payment validation failure.
    - **Next State**: Payment Error Message ("Payment failed! Please try again or use a different payment method.")
- **Transition**: Seat selection error.
    - **Next State**: Error Message ("Oh no! It seems to be an error with your seat selection, please select new seats.")

4.4. Development Plan and Timeline

The development process will be organized into specific categories—Frontend Development, Backend Development, Testing, and Deployment—ensuring that each component is managed efficiently. The following structured approach to developing the Epic Tickets platform, detailing key tasks, team responsibilities, and a clear timeline for project completion.

4.1Task Partitioning

Each major development area is broken down into detailed tasks, allowing for precise tracking and efficient work distribution.

- Frontend Development

    - **Lead:** Efrain Avendano-Gutierrez
    - Design and implement a functional UI utilizing Next.js.
    - Allow for a mobile-responsive application design that also works well with different types of devices.
    - Integrate the front-end components with the back end for data retrieval and submission.
    - Develop components for browsing, payment forms, notifications and selecting seats.
- Backend Development
    - **Lead:** Jay Southwick
    - API endpoints implementation for payment forms and processing using Stripe API
    - Secure theater, movie, and user data management.
    - Set up a database, and backend server utilizing CloudSQL and Java.
    - Detailed API Documentation.
- DevOps
    - **Lead:** Jorge Arellano
    - Infrastructure setup utilizing Google Cloud platform
    - Configure CI/CD pipelines.
    - System performance monitoring.
- Backup handling. Testing
    - Lead: Jorge Arellano
    - Complete unit and integration tests.
    - Conduct User Acceptance Testing (UAT) to make sure requirements are met, and the site is functional.
- Deployment
    - **Lead**: Noorman Amanuel
    - Deploy utilizing Google Cloud Platform.
    - Run optimization checks for performance and site security.
    - Run load test to ensure traffic handling.4.2 Timeline

The project shall follow an 18-week timeline allowing six three-week long sprints.

- Sprint 1:
    - Focus: Requirement and design analysis, and environment setup.
- Output: Finalized SRS documentation, and fully configured development environments. Initial commit.Sprint 2:
    - Focus: Begin stack development, both front and back end.

- Output: Set up a Basic UI, API implementation a design. Database config and front-end component setup.

Sprint 3:

- Focus: Integrate back end with frontend, and development of account management. Initial CI/CD.
- Output: User account management with login and register ability. Profile customization and data storage in a database functional user related API endpoint.

- Sprint 4:
  - Focus: Payment processing implementation and seat selections. Polishing of account management features.
  - Output: Fully functioning payment processing integration, and stable seat selection with a visual interface. Proper implementation of error handling and input validation.
- Sprint 5:
  - Focus: Conduct security penetration test, and user feedback gathering.
  - Output: Performance optimizations, bug fixing and user experience adjustments based on user feedback.
- Sprint 6:
  - Focus: Final testing, quality assurance. Deployment, and finalized project documentation.
  - Run stress test to confirm infrastructure can handle high load.
  - Output: Fully deployed the Epic Tickets system. A comprehensive documentation and report.

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