Movie Ticket Website Assignment

Software Requirements Specification

Final Version (Version 5)

December 11th, 2025

Group 1

Seth V., Christena B., Oscar D.

Prepared for

CS 250, Introduction to Software Systems

Instructor: Gus Hanna, Ph.D.

Fall 2025

Revision History

Date	Description	Author	Comments
09/11/25	Ver. 1	Seth Vanegas	Creation of Document
09/18/25	Ver. 1	Seth Vanegas	First Edits For Sec. 1,2,3
09/18/25	Ver.1	Christena Bell	First Edits For Sec. 1,2,3
09/25/25	Ver. 1	Oscar DeHamer	Last Touches Before Submission
09/25/25	Ver. 1	Seth Vanegas	Last Touches Before Submission
09/25/25	Ver. 1	Christena Bell	Last Touches Before Submission
10/09/25	Ver. 2	Seth Vanegas	Edits for Software Design Specs.
10/09/25	Ver.2	Christena Bell	First edits for Ver.2
10/23/25	Ver. 3	Seth Vanegas	Beginning of Ver. 3
10/23/25	Ver.3	Christena Bell	Beginning of Ver.3
11/04/25	Ver. 4	Seth Vanegas	Beginning of Ver. 4
11/05/25	Ver. 4	Christena Bell	Beginning of ver .4
12/02/25	Ver. 5	Seth Vanegas	Beginning of Ver. 5
12/02/25	Ver. 5	Christena Bell	Beginning of Ver. 5
12/02/25	Ver. 5	Oscar DeHamer	Beginning of Ver. 5
12/11/25	Ver. Fin.	Seth Vanegas	Final correction for final submision

Document Approval

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

Signature	Printed Name	Title	Date
	Seth Vanegas	Computer Scientist	12/11/2025
	Dr. Gus Hanna	Instructor, CS 250	

Table of Contents:

4. Analysis Models

5. Change Management Process

6. Software Design Specifications

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) delineates the scope, features, constraints, and quality attributes of the San Diego Sunset Drive-In Management System (SDSDIMS). It establishes a definitive baseline for the processes of design, implementation, and testing. The stakeholders encompass customers, staff, managers, and developers.

1.2 Scope

The software product to be developed is the San Diego Sunset Drive, In Management System (SDSDIMS). It is a comprehensive system for managing ticketing, parking reservations, concessions, scheduling, and reporting for a retro, style outdoor drive, in theater.

System Capabilities

- Permit customers to browse movies, reserve parking spots, and purchase tickets or concessions either online or on, site.

- Distribute digital tickets featuring QR codes and barcodes to facilitate seamless validation at entry points.

- Authorize personnel to scan and validate tickets, facilitate walk, up sales, and oversee concession inventory.

- Provide managers with tools for scheduling movies, configuring pricing and promotions, and generating operational reports.

The SDSDIMS will not manage external vendor relationships, third, party concession supply orders, or the streaming of movies. Additionally, it will not support functionalities beyond the scope of a single, location drive, in, such as integration across multiple theaters.

The application is customized to support the operations of a drive, in theater featuring a nostalgic “sunset” theme, emphasizing both customer convenience and operational effectiveness.

- **Customer Benefits**: Enable seamless browsing, reservations, and payments to minimize on, site wait times and guarantee the availability of preferred parking.

- **Staff Benefits**: Streamline ticket validation through mobile and POS devices, facilitating efficient tracking of concession sales.

- **Managerial Benefits**: Deliver parameter, driven, user, definable reports (e.g., sales by showtime, parking utilization, concession stock levels) with a turnaround time of less than two hours. Enable online entry of scheduling and pricing parameters.

- **System Goals:** To ensure reliability with a 99.9% uptime, maintain security in compliance with PCI, DSS standards, and support scalability to accommodate up to 2,000 concurrent users during peak events.

By delineating these objectives and constraints, the scope of the SDSDIMS guarantees that the system caters to fundamental operational requirements while circumventing superfluous complexity, aligning with best practices in software engineering for clarity and maintainability.

1.3 Definitions, Acronyms, and Abbreviations

Definitions:

- **Parking Spot Hold** , A temporary reservation of a drive, in parking spot that expires if payment is not completed within a defined time window.

- **Showtime** , A scheduled instance of a movie presentation, defined by movie, start/end time, and assigned parking layout.

- **Digital Ticket (QR/Barcode)** , An electronic code issued upon payment that is scanned for entry validation.

- **Concession Item** , Any food or beverage product sold at the drive, in (e.g., popcorn, soda, candy).

- **Promotion Code** , A rule, based discount or offer applied to a customer's order (e.g., "Buy 2 tickets, get 1 free").

Acronyms:

- **SRS** , Software Requirements Specification

- **SDSDIMS** , San Diego Sunset Drive, In Management System

- **POS** , Point of Sale (cashier/staff terminal for sales)

- **ADA** , Americans with Disabilities Act (U.S. accessibility standard)

- **WCAG** , Web Content Accessibility Guidelines (W3C accessibility standard)

- **PCI, DSS** , Payment Card Industry , Data Security Standard (compliance for payment processing)

- **UI** , User Interface

- NFR , Non, functional requirment

Abbreviations:

- **Mgr.** , Manager

- **Wrk. ,** Worker

- **Cust.** , Customer
- **Sys.** , System
- **Auth.** , Authorized (payment status)
- **Redeem.** , Redeemed (ticket status after scan)

1.4 References

The following documents and materials are referenced in this Software Requirements Specification (SRS) for the **San Diego Sunset Drive, In Management System (SDSDIMS):**

1. **CS250: Introduction to Software Systems , Lecture 1 Notes**
    a. Title: *Lecture 1 , Introduction to Software Systems*
    b. Instructor: Dr. Gus Hanna, San Diego State University
    c. Date: Fall 2025
    d. Source: Provided on Canvas course site
2. **CS250 Assignment Guidelines**
    a. Title: *Requirements Specification Assignment and Rubric*
    b. Report: Assignment posted to Canvas
    c. Date: September 2025
    d. Publishing Organization: San Diego State University, Department of Computer Science
    e. Source: [Canvas course portal]
3. **NATO Software Engineering Conference Report (1968)**
    a. Title: *Software Engineering: Report of a Conference Sponsored by the NATO Science Committee*
    b. Date: October 1968
    c. Publishing Organization: NATO Science Committee
    d. Source: Provided in course readings
4. **Tutorialspoint Software Engineering Tutorial**
    a. Title: *Software Engineering Tutorial*
    b. Report Number: N/A
    c. Date: 2014
    d. Publishing Organization: TutorialsPoint (I) Pvt. Ltd.
    e. Source: Provided in PDF for CS250 course readings
5. **Frederick P. Brooks, Jr.**
    a. Title: *No Silver Bullet, Essence and Accident in Software Engineering*

b. Report Number: IFIP Tenth World Computing Conference Proceedings
c. Date: 1986 (reprinted in *The Mythical Man, Month*, Anniversary Edition, Addison, Wesley, 1995)
d. Publishing Organization: Elsevier Science B.V.
e. Source: Provided in PDF for CS250 course readings

1.5 Overview

This Software Requirements Specification (SRS) delineates the functional and non, functional requirements for the San Diego Sunset Drive, In Management System (SDSDIMS).

The remainder of the document is organized as follows:

- **Section 2: User Requirements:** Provides a comprehensive overview of the operational context, application environment, target platforms, assumptions, and constraints. This section emphasizes the system from the user's perspective, ensuring clarity and professionalism in presentation.

- **Section 3: Functional Requirements:** Defines the system's entities, properties, states, and interrelationships. It describes the expected capabilities in detail, including browsing showtimes, parking spot reservations, ticketing, concessions, scheduling, and reporting. Three representative **use cases** are included to illustrate system behavior from a user standpoint.

- **Section 4: Non, Functional Requirements:** Specifies quality attributes such as performance, usability, security, reliability, maintainability, and compliance. These requirements encapsulate system, wide characteristics that are not associated with individual features but are essential for overall success.

Subsequent assignments in CS250 will expand this Software Requirements Specification (SRS) to encompass design models, testing strategies, and maintenance considerations, thereby building upon the requirements documented herein. This structured approach ensures clarity, adherence to the course SRS template, and traceability for subsequent phases of the project.

2. General Description

This section provides an overview of the environment in which the San Diego Sunset Drive, In Management System (SDSDIMS) will operate, the categories of users expected to interact with the system, the overall functionalities of the product, as well as the assumptions and dependencies influencing the requirements. It does not specify detailed requirements but offers the essential context for their understanding.

2.1 Product Perspective

The San Diego Sunset Drive, In Management System (SDSDIMS) is an innovative, standalone solution developed to facilitate the operations of a single, site drive, in theater. It serves as a contemporary alternative to conventional, manual ticketing and scheduling procedures, consolidating essential business functions, including ticket sales, parking reservations, concession ordering, and reporting, into a unified system.

The system shall not be integrated as part of a larger multi, theater enterprise solution; however, it must be capable of interoperability with external services and technologies, including:

- **Payment gateways (e.g., Stripe, PayPal, or equivalent)** for secure processing of online and on, site transactions.

- **Email and SMS services** are provided for the delivery of digital tickets and customer notifications.

- **Barcode and QR code scanning hardware** employed for the validation of digital tickets at the theater entrance.

- **POS terminals and kiosks** are designated for walk, up sales and concession management.

The SDSDIMS is structured in three tiers:

1. **Customer, Facing Layer** , A responsive web portal and optional mobile application for browsing movies, reserving parking spaces, and purchasing tickets and concessions.

2. **Operational Layer** , Staff, facing point, of, sale applications and scanning tools that facilitate on, site sales and ticket validation.

3. **Management Layer** , An administrative dashboard intended for managerial use to configure schedules, establish pricing structures, develop promotional campaigns, and generate analytical reports.

Although the SDSDIMS functions as an independent product, it leverages best practices from analogous ticketing and entertainment management systems, such as online movie ticketing platforms and live event reservation systems. Its distinctive emphasis is on the drive, in theater experience, which necessitates specific considerations including parking spot reservations, open, air operations, and concession pickup coordination.

The product's perspective aligns with IEEE Standard Requirements Specification (SRS) guidance: it delineates how the system integrates within its operational environment, specifies dependencies on external services, and elucidates the distinctions from related products.

2.2 Product Functions

The San Diego Sunset Drive, In Management System (SDSDIMS) offers comprehensive support for customer engagement, staff management, and executive oversight. At an overarching level, the system is designed to execute the following functions:

Movie and Showtime Management

- Display available movies, formats, and showtimes to customers.
- Allow managers to create, update, and publish showtimes along with assigned parking maps.

Parking Spot Reservation and Ticketing

- Present customers with an interactive parking map displaying real, time availability.
- Permit customers to select parking spaces, place temporary holds, and finalize ticket purchases.
- Issue digital QR/barcode tickets after successful payment.

Payment Processing

- Securely process payments through third, party gateways.
- Support multiple payment methods, including credit card, debit card, and mobile wallet.
- Utilize promotional codes and discounts as established by management.

Concession Ordering and Inventory Tracking

- Allow customers to pre, order concessions via online platforms or to make purchases on, site.

- Permit staff to document sales at point, of, sale terminals.

- Automatically regulate inventory levels and inform managers of low stock.

Ticket Validation and Access Control

- Ensure that personnel are equipped with scanning devices to authenticate digital tickets at the point of entry.

- Monitor redeemed tickets to prevent fraudulent re, use.

- Support offline validation with synchronization when connectivity is restored.

Reporting and Analytics

- Generate reports on ticket sales, parking utilization, concession sales, and promotion usage.

- Provide real, time dashboards for managers to monitor operational performance.

- Export data for financial and administrative purposes.

System Administration and Configuration

- Allow managers to configure user roles (customer, staff, manager).

- Ensure implementation of access control and authentication mechanisms, including multi, factor authentication specifically for managerial personnel.

Support the configuration of pricing tiers, promotions, and concession menus.

2.3 User Characteristics

The San Diego Sunset Drive, In Management System (SDSDIMS) is designed to be accessible to a diverse array of users, most of whom possess limited technical expertise. The system's design prioritizes simplicity, clarity, and usability, ensuring effective interaction for all user groups.

Customers

- **Profile**: The general public, encompassing families, young adults, and older generations.
- **Technical Skills**: Basic familiarity with web browsers, smartphones, and common payment methods.
- **Expectations**: The system offers straightforward navigation, a rapid checkout procedure, and a clear display of movies, showtimes, and parking options.
- **Design Consideration**: Interfaces shall be designed to be intuitive, mobile, compatible, and compliant with ADA/WCAG standards to accommodate a broad range of accessibility requirements.

Staff (Cashiers, Attendants, Ticket Validators)

- **Profile**: Theater personnel responsible for ticket validation, concession sales, and customer support.
- **Technical Skills**: Moderate; trained to operate POS systems and mobile scanning devices but not anticipated to possess extensive technical expertise.
- **Expectations**: Efficient and error, tolerant tools that minimize transaction time and reduce manual entry.
- **Design Consideration**: Staff, oriented tools will employ simple workflows, prominently sized clear buttons, and facilitate offline ticket validation during connectivity disruptions.

Managers/Administrators

- **Profile**: Theater managers are responsible for overseeing scheduling, pricing, promotions, and reporting.

- **Technical Skills**: Superior to customers and staff; skilled in utilizing web dashboards, spreadsheets, and report generation.

- **Expectations**: Access to configuration options, parameter, driven reporting, and secure administrative controls.

- **Design Consideration**: Management dashboards will offer structured workflows, customizable reports, and multi, factor authentication to enhance security.

Overall Principle:

All three user groups will engage with SDSDIMS via role, appropriate, user, friendly interfaces, guaranteeing that the system is accessible, efficient, and conducive to the drive, in theater's nostalgic yet contemporary experience.

2.4 General Constraints

The development and operation of the San Diego Sunset Drive, In Management System (SDSDIMS) are subject to the following general constraints:

Regulatory and Compliance Constraints

- **PCI, DSS Compliance**: All payment transactions are required to comply with the Payment Card Industry Data Security Standards. The system must not retain raw cardholder data.

- **ADA/WCAG Compliance**: Customer, facing applications are required to adhere to accessibility standards as specified by the Americans with Disabilities Act (ADA) and the Web Content Accessibility Guidelines (WCAG 2.1 AA).

- **Privacy Laws**: The system must adhere to relevant privacy regulations, such as the California Consumer Privacy Act (CCPA), including provisions for data access and deletion requests.

Hardware and Platform Constraints

- The system must support standard POS terminals, ticket scanners, and receipt printers already deployed at the theater.

- Customer, facing portals are required to operate on contemporary web browsers such as Chrome, Firefox, Safari, and Edge, as well as on mobile devices, without the necessity for specialized hardware.

- Offline ticket validation must operate on staff mobile devices possessing limited local storage and processing capacities.

Integration Constraints

- Payment processing shall depend on third, party gateways (such as Stripe and PayPal), necessitating compliance with their APIs, transaction limitations, and service accessibility.

- Email and SMS ticket delivery must integrate with external messaging services.

- Management is obliged to manually input showtimes, movies, and promotional content unless a compatible metadata source is adopted.

Performance Constraints

- The availability of parking spots and showtime data must be updated across all channels within two seconds or less in order to prevent double, booking.

- The system must accommodate a peak load of approximately 2,000 concurrent customers without experiencing service degradation.

Security Constraints

- All sensitive data, including user credentials and personal information, must be encrypted both in transit (TLS 1.2 or higher) and at rest (AES, 256).

- Role, based access control must be enforced to prevent unauthorized use of staff or manager tools.

- Manager accounts are required to utilize multi, factor authentication (MFA).

Development Constraints

- The SRS, design, and implementation will adhere to the course guidelines for CS250: Introductory to Software Systems.

- Development will be confined to a single, location deployment, as scalability across the entire chain is not anticipated in this version.

- The design must prioritize maintainability and modularity for future extensions, in line with software engineering best practices.

2.5 Assumptions and Dependencies

Assumptions

- **Internet Connectivity**: It is assumed that dependable internet connectivity will be accessible at the venue for the synchronization of online ticketing, reporting, and concession sales.

- **Customer Device Access**: It is assumed that customers possess access to devices capable of receiving QR or barcode digital tickets, such as email, enabled devices or smartphones.

- **Payment Gateway Availability**: A minimum of one external payment processor (such as Stripe, PayPal) will be accessible and fully functional to facilitate secure financial transactions.

- **Standard Hardware Deployment**: Ticket scanners, receipt printers, and POS terminals are presumed to be available and operational at the theater.

- **Staff Training**: It is assumed that staff and managers will receive adequate training to utilize the POS systems, ticket validation tools, and reporting dashboards.

Dependencies

- **Third, Party Services**: The system relies on external services such as payment gateways, email/SMS ticket delivery, and potentially concession vendor APIs to ensure smooth operation.

- **Regulatory Environment**: The requirements are contingent upon adherence to PCI, DSS, ADA, WCAG, and privacy regulations such as CCPA. Amendments to these standards may necessitate modifications to the system.

- **Movie Metadata**: Accurate movie titles, runtimes, ratings, and promotional content must be provided by theater management or trusted external sources.

3. Specific Requirements

This section delineates the functional and non, functional requirements of the San Diego Sunset Drive, In Management System (SDSDIMS). Each requirement is distinctly identified, verifiable, and prioritized.

3.1 External Interface Requirements

3.1.1 User Interfaces

- **Display all movies currently playing**
  - I. The system will allow the user to see all movies that tickets can be purchased for
  - II. The system will allow the user to add any number of movie tickets to their cart
  - III. The system will display the price of the tickets
- **Reserve parking spaces**
  - I. The system will allow the user to choose a parking space to watch the movie from
  - II. The system will display a map of all available parking spots
  - III. The system will show how much a parking spot costs.
- **Use digital payment system to secure payments**
  - I. The system will allow for the use of Visa and Mastercard
  - II. The system will allow for the purchase of tickets, parking spots, and concessions
- **Generate tickets with unique QR codes**
  - I. The system will generate tickets with unique QR codes that can be validated by staff
- **Login page**
  - I. The system will have a login page that can be accessed by a user's email and a password they create
  - II. The system will save users payment information for quicker transactions
  - III. The system will allow for staff and managers to access systems that customers cannot

- **Scan and validate tickets**
  I. The system will have a QR code scanner that can validate customers tickets
  II. This system will only be able to be used by employees
- **Sell tickets on site**
  I. The system will allow for the sale of tickets on site facilitated by employees
  II. This system will only be able to be used by employees
- **Charts and data dashboard**
  I. The system will allow management to see tickets and other sales data and charts expanding on the data.

3.1.2 Hardware Interfaces

- **Barcode scanner**
  I. Used for validating tickets and purchased concessions by staff to enter the drive, in area
  II. Only used by staff

3.1.3 Software Interfaces

- **Email services**
  I. The system will send emails to users with accounts to tell them about upcoming movies
  II. The system will remind users of tickets they have purchased
- **Payment gateways**
  I. The system will have integration with payment gateways that allow the use of credit cards to pay for tickets, concessions, and parking
- **Database systems**
  I. The system will have a database system to be able to store ticket, parking and concession data
- **Reporting tools**
  I. The system will have compatibility with spreadsheets to be able to allow management users to understand and use data collected

3.1.4 Communications Interfaces

- **Web access**
  I. Allow for web access for all users
- **Mobile access**
  I. All for users to access the product from a mobile device
- **Ticket validation**
  I. The system allows for tickets to validated with a central server

3.2 Functional Requirements

3.2.1 Movie and Showtime Management

I. Introduction

- The system must allow management to update current movies that are playing so customers can browse the movies they can buy tickets for.

II. Inputs

- Mangers will be able to input data about the movies including the title, genre, rating, screen it will play on, start time and end time

III. Processing

- The system will store movies and showtime data in the database and publish the data to customer facing interfaces.

IV. Outputs

- Customers will see the most updated list of movies and their showtimes.

V. Error Handling

- If movie data is missing, it will prompt the manager to input that data before saving it to the database.

3.2.2 Parking Spot Requirements

I. Introduction

- The system must allow customers to reserve their parking spots and purchase tickets for the showtime they choose.

II. Inputs

- Customer can select a showtime
- Customer can see an interactive parking map and select a spot to watch the movie
- Customer proceeds to payment

III.**Processing**

- System will temporarily place a hold on the selected parking spot during checkout
- After the payment goes through, the reservation is finalized and the ticket with an unique barcode is sent to the customer.

IV.**Outputs**

- Confirmation page displaying the reserved parking spot and ticket details
- Digital ticket sent to the customer

V.**Error Handling**

- If payment does not go through, the reservation hold will expire, and the user will be prompted to try again
- If multiple users try to reserve the same spot, whoever finishes their payment first will get the spot.

3.2.3 Payment Processing

I.**Introduction**

- The system must be able to handle financial transactions for tickets, concessions, and parking reservations

II.**Inputs**

- Customer payment details

III.**Processing**

- System will validate the customers payment details
- It will save the transaction in a database

IV.**Outputs**

- Payment confirmation will be sent to the customer via email

V.Error Handling

- If payment is declined, it will prompt the user to try another form of payment

3.2.4 Concession Ordering

I.Introduction

- The system supports concession ordering online and on, site

II.Inputs

- Customers select the concession items that they want

III.Processing

- System reduces inventory levels for the purchased items
- Tells staff which items to prepare before the movie

IV.Outputs

- A receipt that is provided to the customer
- Inventory reports for management

V.Error Handling

- If an item is out of stock, the system will prevent customers from ordering it

3.2.5 Payment Processing

I.Introduction

- The system must enable staff to validate tickets at entry points

II.Inputs

- Customer’s ticket with QR code

- Staff scans ticket with scanner

III. **Processing**

- System checks database for ticket validity
- Marks ticket as used

IV. **Outputs**

- Staff device displays ticket confirmation

V. **Error Handling**

- If ticket is invalid the system alerts the staff

3.2.6 Reporting and Analytics

I. **Introduction**

- The system must provide managers with dashboards that show data about the drive, in

II. **Inputs**

- The manager selects the reports they would like to see, for example ticket sales, concession sales or the most often reserved parking spots.

III. **Processing**

- System will query the database to get the data
- The system will generate data visualizations to be used to understand the data.

IV. **Outputs**

- On, screen dashboards that show the data

V. **Error Handling**

- If requested data is not available, the system will prompt the user to change what they are looking for.

3.3 Use Cases

3.3.1 UC, 1: Reserve Parking and Purchase Tickets

- **Actor(s):** Customer (guest or registered)
- **Purpose:** The customer reserves a parking space and acquires tickets.
- **Preconditions:** Showtimes are available; the payment gateway is operational online.
- **Flow of Events:**

  I. The customer selects a movie and a showtime.

  II. The system displays the available parking spots.

  III. The customer selects one or more spots; the system subsequently places a five, minute hold.

  IV. The customer proceeds to the checkout, inputs payment information, and submits the transaction.

  V. The system verifies payment completion, designates locations as Sold, and issues digital QR/barcode tickets.

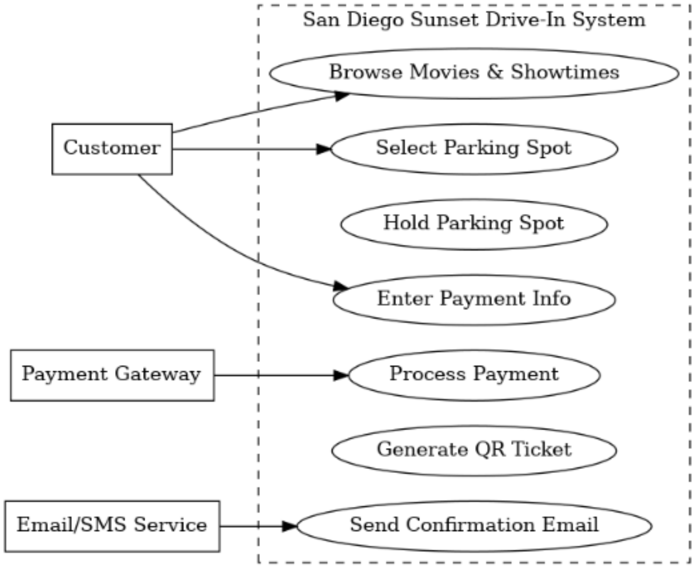
- **Extensions:**

  I. Payment failure → an error message is displayed, and a retry is permitted.

  II. Conflict detected → please reselect promptly.

  III. Hold expires → spots are released into the availability pool.

- **Postconditions:** The ticket and receipt have been dispatched via email; tickets can be accessed in the user's account if registered.



3.3.2 UC, 2: Validate Tickets at Entry

- **Actor(s):** Staff (ticket validator)
- **Purpose:** Personnel verify customer entry into the theater.
- **Preconditions:** The customer possesses a valid QR ticket; the staff device is equipped with scanning capabilities.
- **Flow of Events:**

  I. The customer presents a QR code upon entry.

  II. The personnel member utilizes a validation device to scan the ticket.

  III. The system executes a verification of the ticket status.

IV. If valid and not previously redeemed, the system designates it as Redeemed and grants access to the customer.

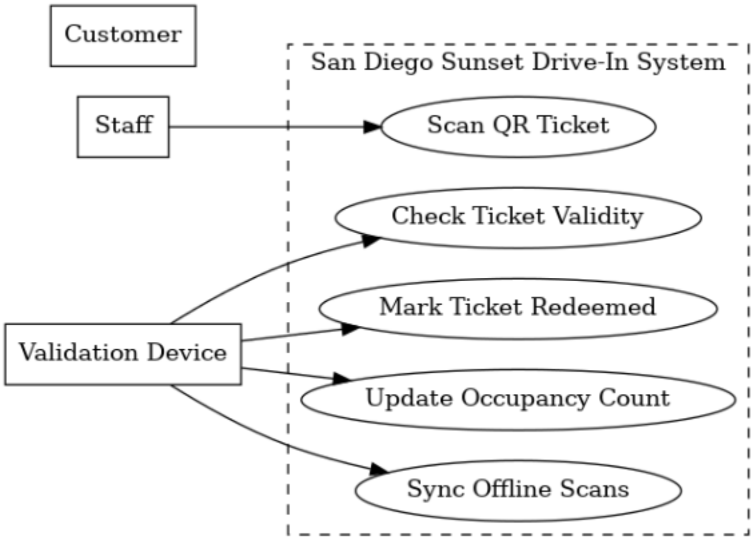
- **Extensions:**

I. Already Redeemed → Any individual who has already been redeemed should be denied entry, and the staff must be duly notified.

II. Invalid ticket → entry will be denied.

III. Offline mode → scan queued locally and synchronize when the connection is restored.

- **Postconditions:** The ticket has been marked as Redeemed; the occupancy count has been subsequently updated.



3.3.2 UC, 3: Manager Publishes Schedule

- **Actor(s):** Manager

- **Purpose:** The manager develops and disseminates a daily or weekly program of movies and showtimes.

- **Preconditions:** The system is configured with movie schedules and parking layouts.

- **Flow of Events:**

  I. The manager opts to select 'Create Schedule' on the dashboard.

  II. The manager allocates movies to specific showtimes and provides parking maps.

  III. The manager establishes the pricing tiers and promotional regulations.

  IV. The system performs validation to identify conflicts such as overlaps and capacity issues.

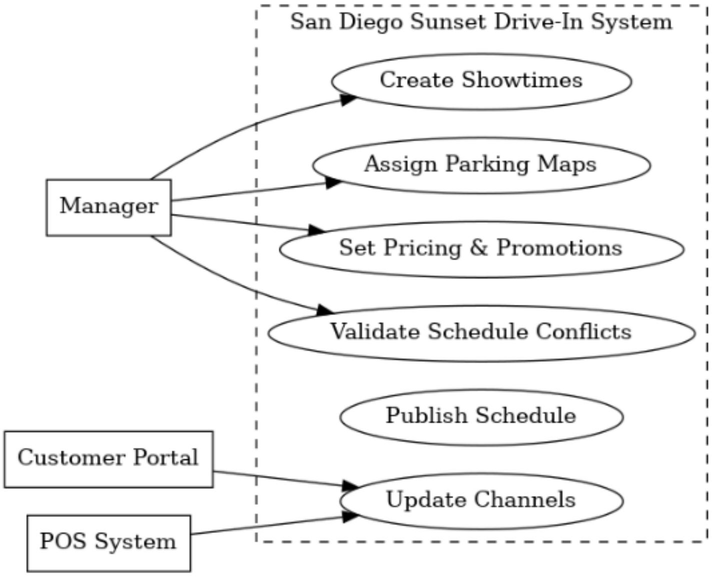
  V. The manager publishes the schedule, enabling customers to browse and purchase tickets.

- **Extensions:**

  I. Time conflict detected → The system prompts the manager to make adjustments.

  II. Pricing error → Please highlight and request correction.

- **Postconditions:** The schedule has been marked as 'On, Sale', and showtimes are available to customers through all channels.



…

3.4 Classes & Objects

This section defines the primary classes and objects of the SDSDIMS, their attributes, and their functions.

3.4.1 Movie

- **Attributes:** movieID, title, duration, rating, genre
- **Functions:** getShowtimes(), addShowtime()
- This relates to the Movie setting functional requirement

3.4.2 Showtime

- **Attributes:** showtimeID, movie, startTime, endTime, parkingLayout
- **Functions:** reserveSpot(), releaseSpot(), sellTicket()
- This relates to the Movie showtime functional requirement

3.4.3 Parking Spots

- **Attributes:** spotID, location, layoutType, isAvailable

- **Functions:** hold(), confirmReservation(), release()
- This relates to the Parking Spot Reservation functional requirement

3.4.4 User

- **Attributes:** userID, name, email, role, paymentInfo
- **Functions:** login(), updateProfile()
- This relates to the Payment Processing functional requirement

3.4.5 Ticket

- **Attributes:** ticketID, showtime, parkingSpot, user, QRcode, status
- **Functions:** validate(), markRedeemed()
- This also relates to the Payment Processing functional requirement

3.4.6 Concession Item

- **Attributes:** itemID, name, price, stockQuantity
- **Functions:** reduceStock(), replenishStock()
- This relates to the Concession Ordering functional requirement

3.5 Non, Functional Requirements

3.5.1 Performance

I. **NFR, P1:** The system shall display movie listings and parking availability within a timeframe of 2 seconds for 95% of requests.

II. **NFR, P2:** Ticket purchase transactions shall be completed (from payment submission to confirmation) within five seconds under normal operating conditions.

III. **NFR, P3:** The system shall accommodate up to 2,000 concurrent customer sessions without any deterioration in response time.

3.5.2 Reliability

I. **NFR, R1:** The system shall possess a Mean Time Between Failures (MTBF) of no less than 30 days.

II.**NFR, R2:** Critical failures, such as payment processing errors and ticket validation system crashes, shall be recoverable within a timeframe of fifteen minutes.

III.**NFR, R3:** Data integrity must be upheld in all transactions; no more than 0.01% of ticketing transactions are permitted to result in data corruption or inconsistency.

3.5.3 Availability

I.**NFR, A1:** The system shall be operational 99.9% of the time within each calendar month, excluding scheduled maintenance periods.

II.**NFR, A2:** Ticket validation devices shall support offline scanning for up to two hours, with synchronization occurring automatically once connectivity is restored.

III.**NFR, A3:** Scheduled maintenance downtime shall not exceed four hours per month and must occur outside peak operating hours.

3.5.4 Security

I.**NFR, S1:** All communications shall be encrypted utilizing TLS 1.2 or a higher version.

II.**NFR, S2:** Sensitive customer data (e.g., PII, order history) shall be stored encrypted at rest using **AES, 256**.

III.**NFR, S3:** Access to staff and manager dashboards shall employ role, based access control (RBAC), with managers mandated to utilize multi, factor authentication (MFA).

IV.**NFR, S4:** The system shall conform to PCI, DSS standards, guaranteeing that no unencrypted credit card data is retained on SDSDIMS servers.

3.5.5 Maintainability

I.**NFR, M1:** A minimum of 70% code coverage shall be maintained through automated testing for business logic.

II.**NFR, M2:** System updates and configuration changes (e.g., pricing, promotions) shall be deployable without requiring system downtime.

III.**NFR, M3:** Documentation shall be updated with each release; changes must be reflected in the SRS/design documents within five business days of deployment.

3.5.6 Portability

I. **NFR, PT1:** Customer, facing applications shall operate on all major browsers, including Chrome, Firefox, Safari, and Edge, within their most recent two versions.

II. **NFR, PT2:** Mobile access shall be supported on devices operating with iOS 14 or later and Android 10 or later.

III. **NFR, PT3:** Staff and manager dashboards shall be accessible on both Windows 10 and later versions, as well as macOS 12 and later versions.

IV. **NFR, PT4:** Migration to a new hosting environment, whether cloud, based or on, premises, should necessitate no more than two weeks of effort, with no modifications to the business logic.

3.6 Inverse Requirements

No inverse requirements were identified for this system.

3.7 Design Constraints

No additional design constraints beyond those stated in earlier sections apply to this system.

3.8 Logical Database Requirements

Logical database requirements are addressed in Section 7: Data Management Strategy.

3.9 Other Requirements

No further requirements were identified outside those previously documented.

4. Analysis Models

Formal analysis models such as sequence diagrams, DFDs, and state machines were not required for this project and are therefore omitted.

4.1 Sequence Diagrams

This model was not required as part of the project deliverables.

4.3 Data Flow Diagrams (DFD)

This model was not required as part of the project deliverables.

4.2 State, Transition Diagrams (STD)

This model was not required as part of the project deliverables.

5. Change Management Process

Changes to the SRS are managed through a controlled review and approval process to ensure consistency and traceability across all project documents. Any team member may request a change by documenting the modification, rationale, and affected sections. Proposed changes are first reviewed collaboratively by the team, then integrated only after consensus is reached. Once approved, updates are added to the SRS and reflected in the corresponding design and architecture sections. Versioning through GitHub ensures that all revisions are tracked, allowing the team to maintain a clear history of modifications throughout the project life cycle.

6. Software Design Specifications

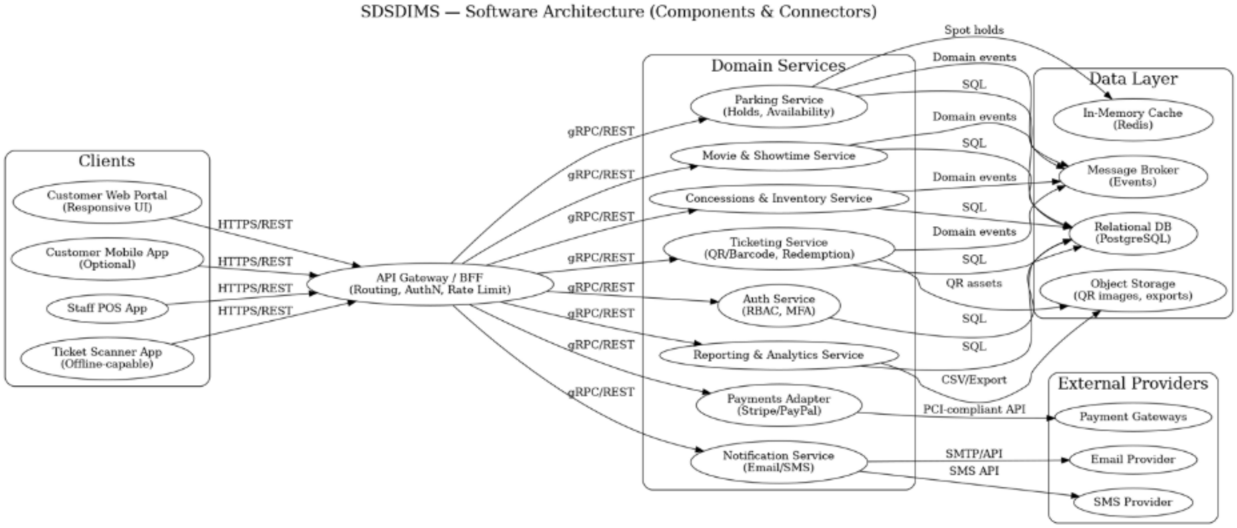
6.1 System Description

The San Diego Sunset Drive, In Management System (SDSDIMS) is a three, tier web platform that supports all aspects of drive, in operations: public movie browsing and ticket purchase; staff tools for on, site sales and ticket validation; and a manager dashboard for scheduling, pricing/promotions, concessions, and reporting. Customers can browse showtimes, select a parking spot on an availability map, and buy tickets and concessions via a secure, mobile, friendly interface. Staff utilize a POS system and a scanner application (which can operate offline) to validate QR/barcode tickets at the entrance and manage walk, up sales. Managers publish schedules, oversee inventory, and review analytics on sales, parking occupancy, and promotional effectiveness.

Architecturally, the system employs a single API Gateway (BFF) to serve all clients. Core domain services, such as Movie & Showtime, Parking (with short, term “hold” logic), Ticketing (QR code generation and validation), Concessions & Inventory, Payments (integrated with Stripe/PayPal), Notifications (Email/SMS), Reporting/Analytics, and Authentication (RBAC+MFA), maintain state within a relational database and utilize a cache for operations with high contention (such as spot holds). A message broker transmits domain events (e.g., TicketPurchased, SpotReleased) to facilitate workflow decoupling and support analytics. All payment transactions are processed through PCI, compliant processors; no raw PAN data is stored within SDSDIMS.

6.2 Software Architecture Overview

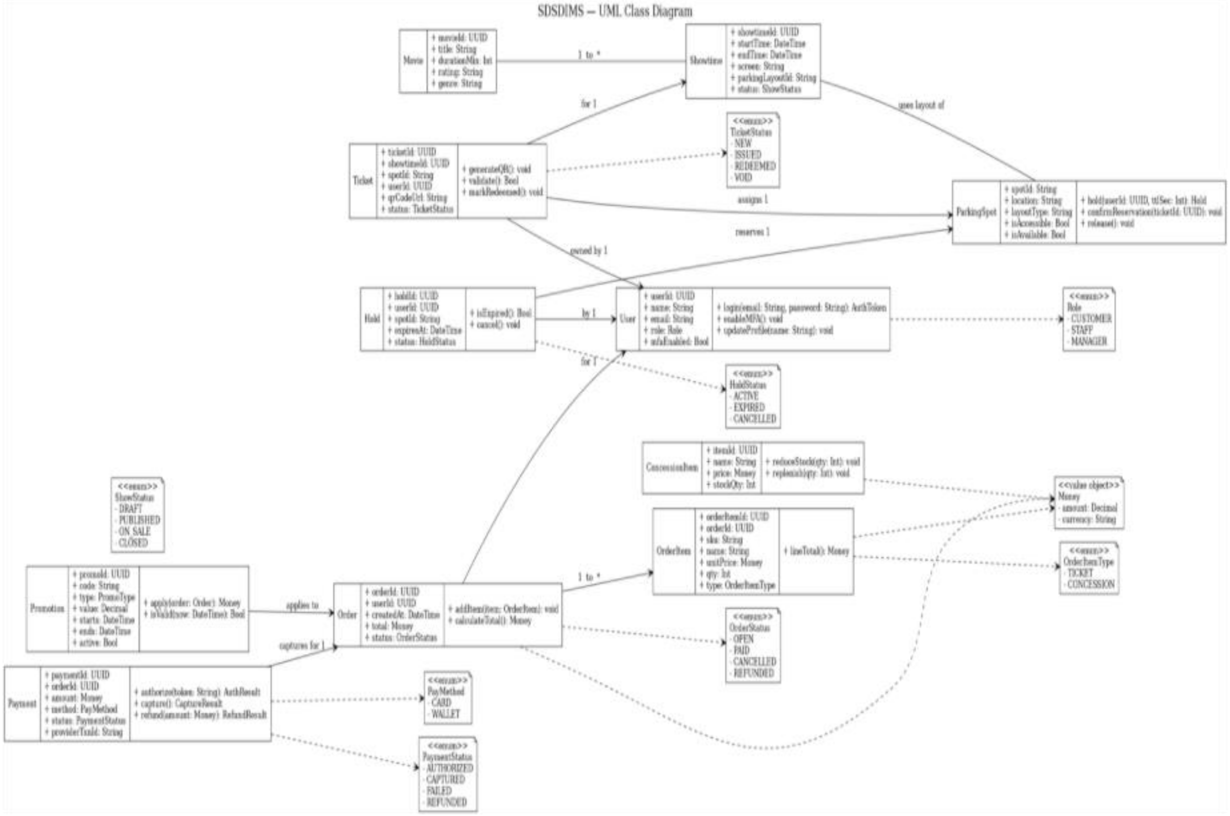
6.2.1 Architecture Diagram


SDSDIMS — Software Architecture (Components & Connectors)
Clients
Customer Web Portal (Responsive UI)
Customer Mobile App (Optional)
Staff POS App
Ticket Scanner App (Offline-capable)
HTTPS/REST
HTTPS/REST
HTTPS/REST
HTTPS/REST
API Gateway / BFF (Routing, AuthN, Rate Limit)
gRPC/REST
gRPC/REST
gRPC/REST
gRPC/REST
gRPC/REST
gRPC/REST
gRPC/REST
gRPC/REST
Domain Services
Parking Service (Holds, Availability)
Movie & Showtime Service
Concessions & Inventory Service
Ticketing Service (QR/Barcode, Redemption)
Auth Service (RBAC, MFA)
Reporting & Analytics Service
Payments Adapter (Stripe/PayPal)
Notification Service (Email/SMS)
Spot holds
Domain events
SQL
Domain events
SQL
Domain events
SQL
Domain events
SQL
QR assets
SQL
SQL
CSV/Export
PCI-compliant API
SMTP/API
SMS API
Data Layer
In-Memory Cache (Redis)
Message Broker (Events)
Relational DB (PostgreSQL)
Object Storage (QR images, exports)
External Providers
Payment Gateways
Email Provider
SMS Provider

6.2.2 Architecture Explanation

Clients (Customer Web, optional Mobile, Staff POS, and Scanner) access the API Gateway over HTTPS. The gateway manages routing, authentication, and rate limiting, and forwards requests to domain services via REST/gRPC. Authentication handles RBAC and MFA; Movie & Showtime services publish schedules; Parking manages availability and time, boxed holds backed by a cache; Ticketing issues and redeems QR/Barcode tickets, storing QR assets in object storage; Concessions and Inventory record on, site and pre, orders; Payments integrate with external gateways; Notifications send email and SMS messages; Reporting aggregates events for dashboards and exports. Services persist data to a relational database and publish domain events to a broker to maintain decoupling (e.g., a purchase triggers notifications and inventory updates). External providers (payments, email, SMS) are accessed via provider APIs.

6.3 UML Class Diagram



6.4 Class Descriptions

The SDSDIMS domain model centers on users purchasing tickets and concessions for scheduled movies. A User logs in, manages a profile, and holds a role that determines access (Customer, Staff, or Manager). Each Movie has multiple Showtimes, which define times and parking layouts. ParkingSpots can be held or reserved through Hold objects, then converted into Tickets containing QR codes for entry validation. Orders group tickets and concessions together, linked to a Payment record and optional Promotion discount. ConcessionItems maintain stock quantities and prices for food and drink sales. Relationships among these classes ensure that each transaction, from reservation to redemption, is securely tracked and easy for both staff and management to manage.

6.5 Development Plan and Timeline

The San Diego Sunset Drive, In Management System will be developed through three principal phases: setup, implementation, and testing/deployment. Responsibilities will be distributed among all team members to guarantee equitable contributions and detectable GitHub commits from each participant.

Phase 1 , System Setup (Week 1):
Seth V. initializes the GitHub repository, establishes the foundational project structure, and integrates essential frameworks for the web portal and API gateway. Christena B. prepares initial HTML/CSS templates and configures page routing. Oscar D. sets up the relational database schema, defining tables for users, movies, and showtimes.

Phase 2 , Core Implementation (Weeks 2, 3):
Seth V. is responsible for implementing authentication, movie management, and parking reservation APIs. Christena B. facilitates the integration of front, end interfaces for browsing movies, processing checkouts, and displaying tickets, ensuring they are responsive. Oscar D. develops the ticket validation module, concession sales interface, and reporting endpoints. All team members perform local testing of their respective modules and submit updates on a weekly basis.

Phase 3 , Integration and Testing (Weeks 4, 5):
The team collaboratively conducts integration testing of payment processes, QR code validation, and reporting accuracy. Seth manages GitHub issue tracking and pull requests; Christena supervises UI refinement and accessibility assessments; Oscar is responsible for database optimization and deployment packaging. Upon verification, the final version of the consolidated SRS and SDS documents, along with architecture and UML diagrams, is uploaded to GitHub, with each team member confirming at least one commit.

Deliverables will be integrated via peer, reviewed pull requests to ensure traceability and uphold quality standards. This phased schedule guarantees consistent progress, code cohesion, and transparent collaboration documentation for evaluation purposes.

7. Test Plan

This section delineates the verification and validation strategy for the San Diego Sunset Drive, In Management System (SDSDIMS). The objective of testing is to ensure that all functional and non, functional requirements specified in this Software Requirements Specification are met, and that the implemented system operates reliably under real, world conditions. Testing shall be conducted iteratively throughout the development process, advancing from isolated module testing to comprehensive system validation.

7.1 Testing Objectives

Testing will verify that the system correctly supports movie scheduling, parking reservations, ticketing, concessions, and reporting functions while meeting security and performance requirements. Validation confirms that the delivered system fulfills stakeholder expectations for usability, responsiveness, and accuracy.

7.2 Test Strategy

Testing will be conducted across three granularities:

I. Unit Testing (Level 1): Individual modules, including the ParkingSpot class, Ticket class, and Payment adapter, will be tested in isolation utilizing automated frameworks. Each function shall be verified against expected outputs, such as successful QR code generation, inventory updates, and payment status modifications.

II. Integration Testing (Level 2): Interactions between primary subsystems shall be assessed, specifically, the checkout process linking the Parking Service, Ticketing Service, and Payments adapter. These evaluations verify that data is transmitted accurately via APIs and that the database preserves consistency throughout multi, step transactions.

III. System Testing (Level 3): End, to, end user scenarios will be executed on the deployed environment, covering both customer and staff workflows. System tests will validate end, user functions such as browsing movies, purchasing tickets, scanning QR codes at entry, and generating managerial reports.

7.3 Testing Environment

Testing will be conducted within a staging environment that closely replicates the hardware and network configurations of the production setting. The database will utilize anonymized sample data. Automated testing will be executed via a continuous integration pipeline, while manual exploratory testing will be employed to verify user interface behavior.

7.3 Test Case Samples

Ten representative test cases will be documented in the accompanying *testCasesSample.xlsx* file and uploaded to the group GitHub repository. These include at least two cases per test granularity:

- Unit tests (e.g., validateTicket(), holdSpot())
- Integration tests (e.g., checkoutWorkflow, paymentGatewayAPI)
- System tests (e.g., customerPurchaseFlow, managerReportGeneration)

Each test case specifies an ID, feature under test, inputs, expected results, and pass/fail criteria.

7.4 GitHub Link

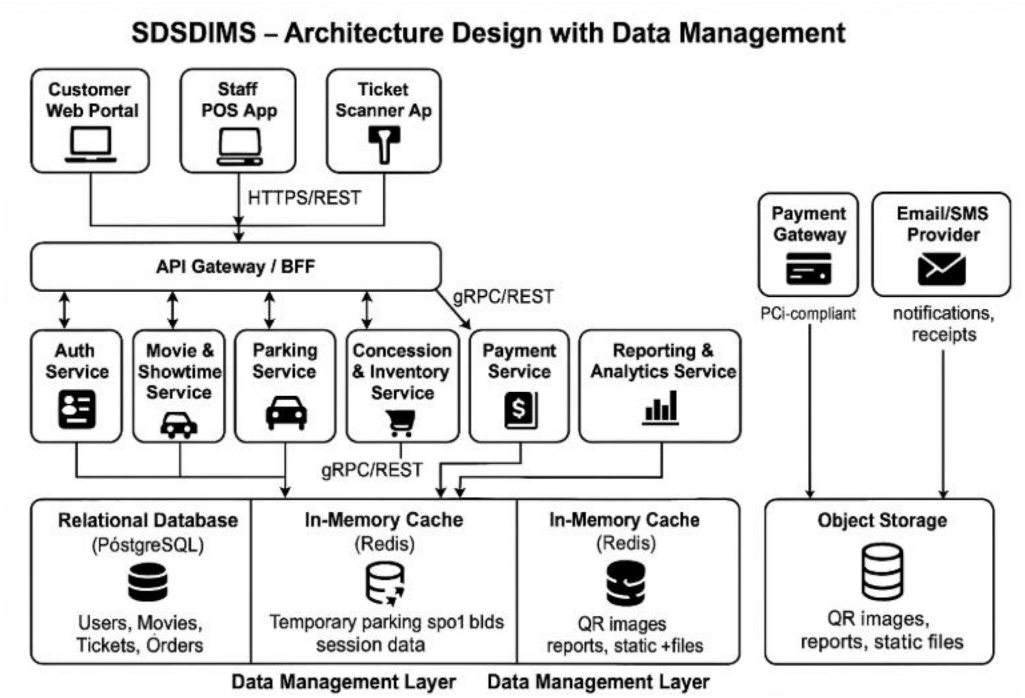
The detailed Test Plan files, including the testCasesSample.xlsx spreadsheet and the Test Case Summary PDF, are available in the project repository:

Redacted-Information/CS250-GROUP1-SRS: Software Requirements & Design Specification for the San Diego Sunset Drive-In Management System (SDSDIMS).

8. Architecture Design with Data Management

8.1 Updated Architecture Design

The architecture of the San Diego Sunset Drive-In Management System (SDSDIMS) incorporates a centralized Relational Database Management System (RDBMS), specifically PostgreSQL, which functions as the enduring data repository for all domain services. This database maintains entities including User, Movie, Showtime, Ticket, ParkingSpot, Order, and Payment. Each microservice communicates with the RDBMS through secure API interactions and adheres to the CRUD (Create, Read, Update, Delete) model. Additionally, Redis-based caching is employed for time-sensitive data, such as temporary parking spot reservations.



8.2 Data Management Strategy

SDSDIMS employs a Relational Database Model to uphold consistency and integrity across its structured data. Tables are normalized to Third Normal Form (3NF) to prevent redundancy and to ensure precise relationships between entities. Each principal subsystem interacts with the database through service APIs that enforce business logic prior to committing changes to the database.

Core Tables include:

- **Users** - stores user accounts, roles, and credentials (with hashed passwords).
- **Movies** - film title, runtime, rating, genre.
- **Showtimes** - associated movie, start/end time, screen, and capacity.
- **ParkingSpots** - location, availability, accessibility.
- **Tickets** - linked to user, showtime, and parking spot; includes QR code path and redemption state.
- **Orders & Payments** - track customer purchases, payment confirmations, and refunds.
- **Concessions** - product catalog, prices, and inventory quantities.

DBMS Features Implemented:

- **Data Integrity:** enforced by primary and foreign keys; referential integrity maintained between related tables.
- **Security:** user authentication via hashed passwords and role-based access control (RBAC).
- **Backup and Recovery:** nightly backups and transaction logs ensure recoverability.
- **Concurrency Control:** two-phase locking ensures consistent updates when multiple users reserve spots simultaneously.
- **Replication:** read replicas may be deployed for analytics and reports.

8.3 Tradeoff Discussion

The relational model was selected due to its adherence to ACID principles (Atomicity, Consistency, Isolation, Durability), which guarantee dependable ticketing and financial transactions. Although NoSQL systems provide enhanced scalability and flexibility, they often sacrifice strict consistency, thereby rendering them less suitable for real-time payment and reservation processes. Nonetheless, the use of caching (Redis) and horizontal scaling of the API gateway will serve to alleviate any performance constraints associated with SQL-based storage solutions.

Alternative architectures utilizing NoSQL technologies were examined for the analytics module; however, the final design favors a hybrid approach: employing relational databases for operational tasks and potentially leveraging NoSQL or data warehouse solutions for long-term reporting.

9. Life-Cycle Model

9.1 Selected Development Model

The San Diego Sunset Drive-In Management System (SDSDIMS) was developed utilizing an Incremental and Iterative Life-Cycle Model. Development proceeded through clearly delineated phases, Requirements, Design, Test Planning, Architecture, and Data Management, each successive phase building upon the preceding deliverable. Although the sequence of tasks resembled a structured Waterfall model, the team consistently refined earlier work in response to feedback from the instructor and TA, a practice characteristic of an Iterative methodology.

This hybrid approach enabled the team to maintain well-defined documentation milestones while enhancing diagrams, requirements, and design elements throughout the project.

9.2 Team Collaboration and Workflow

Team collaboration was primarily coordinated through:

I. Weekly meetings (during class and after class).
II. Group chat communication for urgent questions.
III. Shared Google Docs / Word documents for drafting text
IV. GitHub is utilized for storing diagrams, test plans, and final artifacts.
V. Division of responsibilities, with each member handling specific sections of the SRS, SDS, and Test Plan.

The work was executed in iterative cycles. Each team member independently drafted content, after which the group discussed revisions, consistency, and integration into the final, unified document. Upon receiving feedback from the TA, the team collaboratively updated diagrams, clarified requirements, and addressed formatting issues.

9.3 Challenges Encountered

Several challenges influenced the team's workflow:

I. Ensuring uniformity throughout a lengthy, multi-section document necessitated frequent cross-referencing.
II. Ensuring accuracy in UML and architecture diagrams necessitated multiple iterations.
III. Coordinating schedules to combine work and generate a cohesive report required additional planning.
IV. Gaining an understanding of new concepts such as Database Management System (DBMS) strategies and test case structuring necessitated a supplementary review of the lecture material.

Despite these challenges, the team demonstrated swift adaptation by allocating tasks based on individual strengths and revising their work as necessary.

9.4 Strengths of the Process

I. The incremental structure allowed the team to focus on one component of the system at a time, reducing cognitive load.
II. Feedback cycles improved the quality of the final deliverables.
III. GitHub ensures version control and transparency in contributions.
IV. Collaborative drafting allowed team members to learn from each other's approaches.

9.5 Opportunities for Improvements

If given more time, the team would improve by:

I. Establishing a more formal sprint-based Agile workflow
II. Using issue tracking or a Kanban board to assign tasks
III. Conducting peer reviews before submitting each milestone
IV. Holding additional design review meetings to finalize diagrams earlier

10. Summary

The San Diego Sunset Drive-In Management System (SDSDIMS) is a comprehensive, web-based platform engineered to modernize the operations of drive-in movie theaters. It offers patrons an efficient interface for discovering films, selecting showtimes, reserving parking spaces, purchasing tickets, and gaining entry to the venue via QR code-based digital verification. For staff and management personnel, SDSDIMS provides secure resources for ticket scanning, managing concession inventory, and generating detailed sales and attendance reports.

The system architecture adopts a modular, service-oriented design. An API Gateway manages routing, authentication, and request validation, directing traffic to specialized domain services such as Movie & Showtime, Parking, Ticketing, Concession, Reporting, and Payments. These services interface with a centralized relational database (PostgreSQL) that maintains structured data, including information on users, tickets, movies, orders, and payments. Performance-critical data, including parking holds and session data, is handled via an in-memory caching solution. Digital assets, including QR codes and exported reports, are stored in an object storage service to isolate static content from operational data.

Data management prioritizes consistency, accuracy, and recoverability. SDSDIMS utilizes ACID-compliant transactions, foreign key constraints, hashed credentials, and validated input checks to guarantee the reliability and security of stored data. Nightly backups and transaction logs serve as durable recovery options, while access control and encryption mechanisms safeguard confidential customer and payment information. Stringent validation rules and regulated data reinforce system integrity throughout all services.

A comprehensive Testing Plan verifies system correctness at multiple levels. Unit tests validate the behavior of core classes and methods such as holds, tickets, and stock reductions. Integration tests ensure correctness across service boundaries, including checkout flow, payment processing, and ticket redemption. System tests evaluate full end-to-end scenarios, confirming accurate performance under real usage conditions. Together, these tests support the system’s reliability before deployment.

Overall, SDSDIMS offers a cohesive, scalable, and secure platform that enhances the customer experience, facilitates operational efficiency, and establishes a robust foundation for future developments, including automated demand forecasting, multi-location support, and kiosk-based check-in.

11. Conclusion

The development of the San Diego Sunset Drive-In Management System offered an opportunity to implement core software engineering principles across the entire system life cycle. Beginning with requirements elicitation and progressing through phases such as design, architecture, data management, and testing, this project demonstrated how intricate systems are developed through structured planning and iterative refinement. Each stage necessitated translating real-world operational needs into technical specifications, diagrams, and workflows, which together form a coherent and functional system design.

One of the strengths of the project lies in its comprehensive documentation. Each component, requirements, system architecture, database strategy, and testing plan, was developed with consistency and clarity, ensuring that stakeholders can understand both the high-level goals and the detailed mechanics of the system. The incremental approach also allowed continuous improvement, enabling design corrections and refinements as new insights emerged. The final architecture balances modularity, scalability, and data integrity, providing a solid foundation for implementing a production-ready version of SDSDIMS.

The project simultaneously underscored several prevalent challenges in software development. Coordinating efforts among multiple contributors necessitated deliberate communication and meticulous version control. Ensuring consistency between diagrams and textual descriptions required multiple revisions. Comprehending database normalization, ACID properties, and multi-layer system design entailed revisiting lecture materials and external resources. These challenges ultimately contributed to the enhancement of the final product by fostering a more profound engagement with the fundamental concepts.

Overall, this project enhanced my understanding of the processes involved in the design, documentation, and validation of software systems. It highlighted the importance of well-defined requirements, modular architecture, and exhaustive testing. Most notably, it demonstrated how theoretical principles learned in class are applied to practical design decisions in real-world applications. With additional time, potential future improvements could include the development of a functional prototype, the integration of real-time analytics, and the expansion of support for multi-theater operations. The insights gained from this project will serve as a valuable foundation for my future efforts in software engineering and system architecture.

A.1 User Manual

Hardware Requirements

- A desktop or laptop computer equipped with a minimum 1.8 GHz dual-core processor.
- Four gigabytes of RAM are required, with eight gigabytes recommended.
- At least two gigabytes of available storage space are necessary.
- Mobile devices (iOS/Android) should support modern browsers for QR ticket access.

Software Requirements

- Operating System: Windows 10 or later, macOS 11 or later, or Linux.
- Web Browser: Chrome, Firefox, Safari, or Edge (latest two versions recommended).
- PDF reader for ticket and report downloads.
- Internet connection with minimum 5 Mbps downstream bandwidth.

Network Requirements

- Secure internet connectivity through Wi-Fi or LTE/5G networks.
- HTTPS support for secure communication.
- Port 443 (HTTPS) has been enabled to facilitate browser access.

User Credentials

- Customer users are required to register using their email address and a password.
- Staff users are required to be assigned roles such as Cashier, Parking Attendant, or Manager by an authorized administrator.

A.2 System Overview

The San Diego Sunset Drive-In Management System (SDSDIMS) offers clients and personnel secure access to film schedules, parking reservations, ticket purchases, QR code-based entry validation, and reporting functionalities. The system functions entirely via a web-based interface accessible from both desktop and mobile platforms.

A.3 Customer User Interactions

A.3.1 Account Registration and Login

I. Navigate to the SDSDIMS home page.
II. Please select 'Create Account'.
III. Please provide the necessary information: your full name, email address, and password.
IV. Please select the register to complete the process.
V. For returning users, please select 'Login' and input your email and password.

A.3.2 Browsing Movies

I. Upon logging in, please select 'Movies' from the top navigation menu.
II. The system exhibits all films presently playing and those scheduled for upcoming screenings.
III. Please select a movie poster to view its synopsis, rating, runtime, and showtimes.

A.3.3 Selecting a Showtime

I. Within the movie detail page, scroll to the **Showtimes** section.
II. Each available showtime displays the date, start time, and remaining parking availability.
III. Please select the preferred showtime to proceed to the reservation page.

A.3.4 Parking Spot Reservation

I. The system presents a graphical depiction of available parking spaces.
II. Available spots are indicated in green; unavailable spots are shown in gray.
III. Please select an available spot to place a temporary hold.
IV. The system reserves the position for a duration of five minutes, as indicated by a countdown timer.
V. Please select 'Continue to Checkout' to proceed.

A.3.5 Ticket Purchase and Payment

I. Please review the following summary: movie title, showtime, parking spot, and total price.
II. Please enter your payment information in the secure checkout form.
III. Please select 'Submit Payment'.
IV. Upon successful payment.
   a. The system issues a digital ticket.
   b. A QR code is generated and stored within the user's profile.
   c. A confirmation email has been dispatched.

A.3.6 Accessing and Using the QR Ticket

I. Please navigate to 'My Tickets' from the navigation menu.
II. Please select the preferred showtime to view the QR code.
III. Upon arrival at the drive-in, please present the QR code to the parking attendant.
IV. Staff members utilize the SDSDIMS Scanner App to scan the QR code.
V. The system verifies the ticket and designates it as Redeemed.

A.4 Staff User Interaction

A.4.1 Staff Login

I. Staff members authenticate their accounts using designated credentials.
II. The system automatically loads the Staff Dashboard in accordance with the user's designated role.

A.4.2 Ticket Validation (Parking Attendant)

I. Please open the Scanner Application on the staff device.
II. Please select the scan ticket.
III. Align the device camera with the customer's QR code.
IV. The system displays:
   a. Ticket status (Valid, Invalid, or Redeemed)
   b. Movie and showtime details
   c. Assigned parking spot
V. If valid, please direct the customer to their designated location.

A.4.3 POS Ticket Sales (Cashier)

I. Please access the Ticketing Interface.
II. Locate the preferred film and screening time.
III. Select an available parking spot on behalf of the customer.
IV. Proceed with the payment process via the staff Point of Sale (POS).
V. Provide the customer with printed or digital confirmation.

A.4.4 Manager Reporting

1. Navigate to **Manager Dashboard**.
2. Select **Reports** to view:
   a. Daily sales

   b. Showtimes and attendance
   c. Parking occupancy levels
   d. Concession inventory levels

3. Export reports as **CSV** or **PDF** if needed.