

Table Of Content

1. Reply to the Customer:	1
2. Problem Statement	2
3. MVP Backlog and Acceptance Criteria with Corner Cases	2
User Story 1: Subscribe to Cinema Plan	2
User Story 2: Check-in for Movie Visit	2
User Story 3: Manage Subscription	3
User Story 4: Cross-Platform Consistency	3
4. Timeline (Scrum-Oriented)	4
Sprint 1 (Week 1-2): Discovery Phase and Initial Development	4
Sprint 2 (Week 3-4): Focus on MVP Development	4
Sprint 3 (Week 3-4): MVP Development Begins	5
Sprint 4 (Week 5): Continue MVP Development	5
Sprint 5 (Week 6): Finalize MVP	6
Strengths of this Approach:	6
Weaknesses to Address:	7
5. Security and Payment Enhancements	8

1. Reply to the Customer:

Dear CPO,

Thank you for sharing your insightful feedback and presenting the exciting opportunity to collaborate further.

We've reviewed your request and are prepared to move forward with the cinema subscription service's design and development. To ensure we align on expectations and deliver an MVP that's scalable and efficient, we propose starting with a discovery phase to finalize key details such as integration, security, and payment flow.

Points for our upcoming discussion:

- Detailed subscription models (pricing, limits, and special terms)
- Technical requirements for integration with partner cinemas' systems
- User flow for subscription and ticket booking
- Recurring billing and security considerations (including PCI DSS compliance)
- Reporting and analytics requirements

I'll reach out shortly to arrange a discovery call.

Best regards,
Kacper Gierycz
Project Technical Lead

2. Problem Statement

Customer seeks to introduce a subscription service for partner cinemas within their existing payment app. The service should allow users to subscribe to a cinema and attend movies up to twice a week. This needs to be delivered promptly to drive user engagement and growth while ensuring scalability, security, and seamless integration.

Key Assumptions:

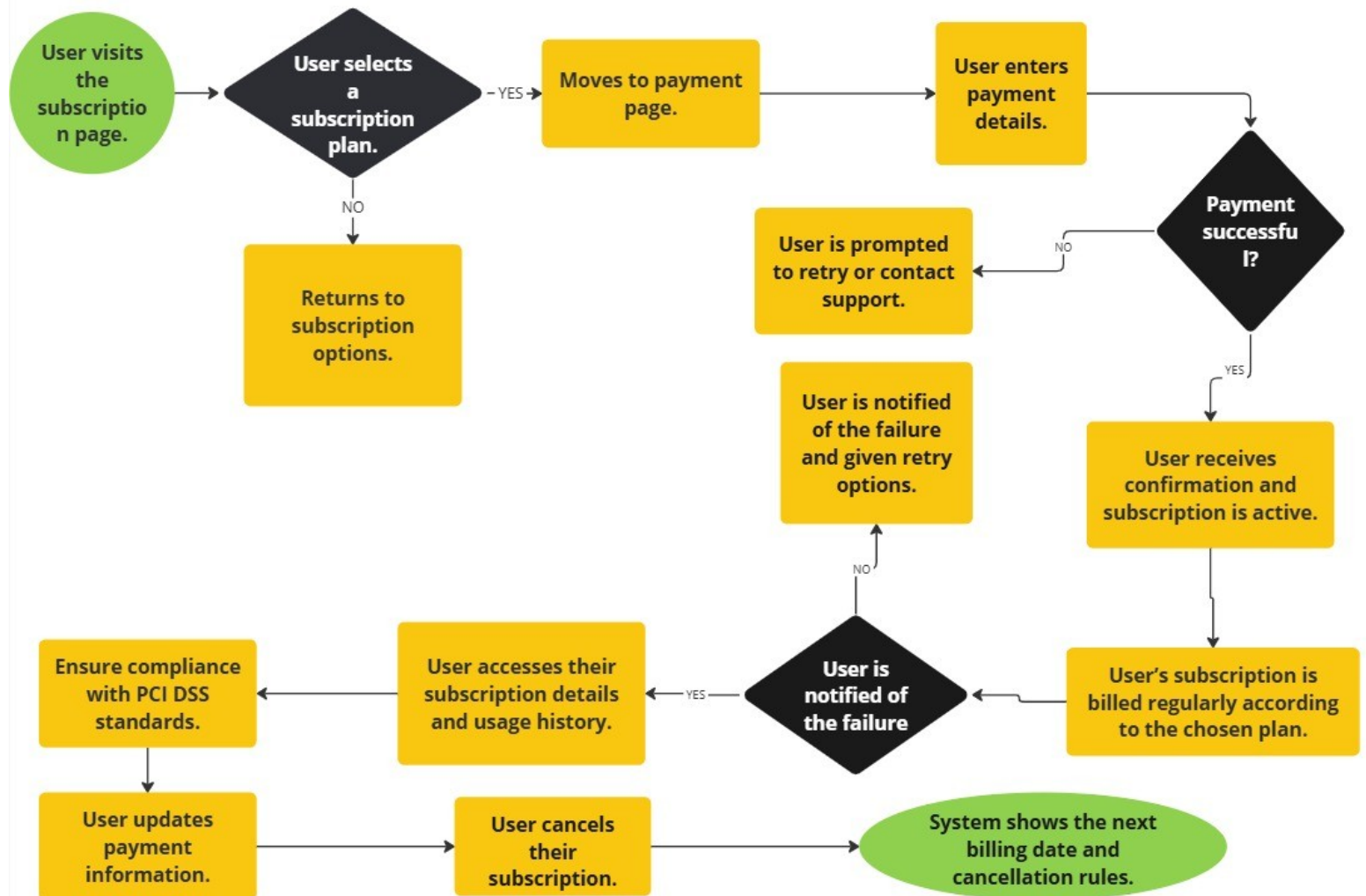
- *Interview all Company stakeholders, for feedback and possible insight before contacting Customer.*
 - *Brainstorm the team for estimated timeline and risks for this project.*
 - *Prepare more detailed roadmap and setup the meeting with the Customer.*
 - Integration with partner cinemas via their booking and payment systems.
 - Subscription management within Customer's app.
 - Secure recurring billing for subscription plans (adhering to PCI DSS).
 - Cross-platform support (iOS/Android).
-

3. MVP Backlog and Acceptance Criteria with Corner Cases

User Story 1: *Subscribe to Cinema Plan for Booking Seats*

As a user **I want to** subscribe to a cinema plan **So that** I can book seats in the cinema with it

Subscription and user details management



Acceptance Criteria:

1. View and Choose Subscription Plans.
 - **Given** the user is viewing subscription plans
 - **Then** they can view and choose from available plans
 - **And** see details such as price, availability, and benefits
2. Complete Subscription Process.

- **When** the user completes the subscription process
- **Then** they can use a secure payment gateway supporting multiple options like cards with PayPal or Stripe
- **And** data transfer between the website and API is secured with SSL or TLS

3. Secure Recurring Billing

- **Given** the user subscribes
- **Then** the payment methods comply with PCI DSS requirements for secure recurring billing

4. Handle Payment Failures

- **When** a payment failure occurs
- **Then** the user is notified and given retry options

5. Upgrade or Downgrade Subscription

- **When** the user wants to upgrade or downgrade mid-cycle
- **Then** they can change their plan, and pro-rata adjustments are applied

6. Retry Failed Card Charge Attempts

- **Given** a failed card charge attempt
- **Then** the system retries in 24 hours and notifies the user

7. Subscription Success Confirmation

- **When** the subscription is successful
- **Then** the user receives confirmation via email and in-app notification

Corner Cases:

1. **Subscription Expiry:** If a user's subscription expires mid-transaction, prompt them to renew or provide alternative payment options.
2. **Multiple Payment Failures:** Notify the user after consecutive payment failures and possibly suspend the service until resolved.
3. **Subscription Overlap:** Handle scenarios where the user tries to subscribe again without knowing they're already subscribed, with clear messages.
4. **Service Downtime:** Provide fallback options or clear messages if the subscription service or payment gateway is down.
5. **Partial Payments:** Handle scenarios where users might partially complete payments due to connectivity issues or interruptions.

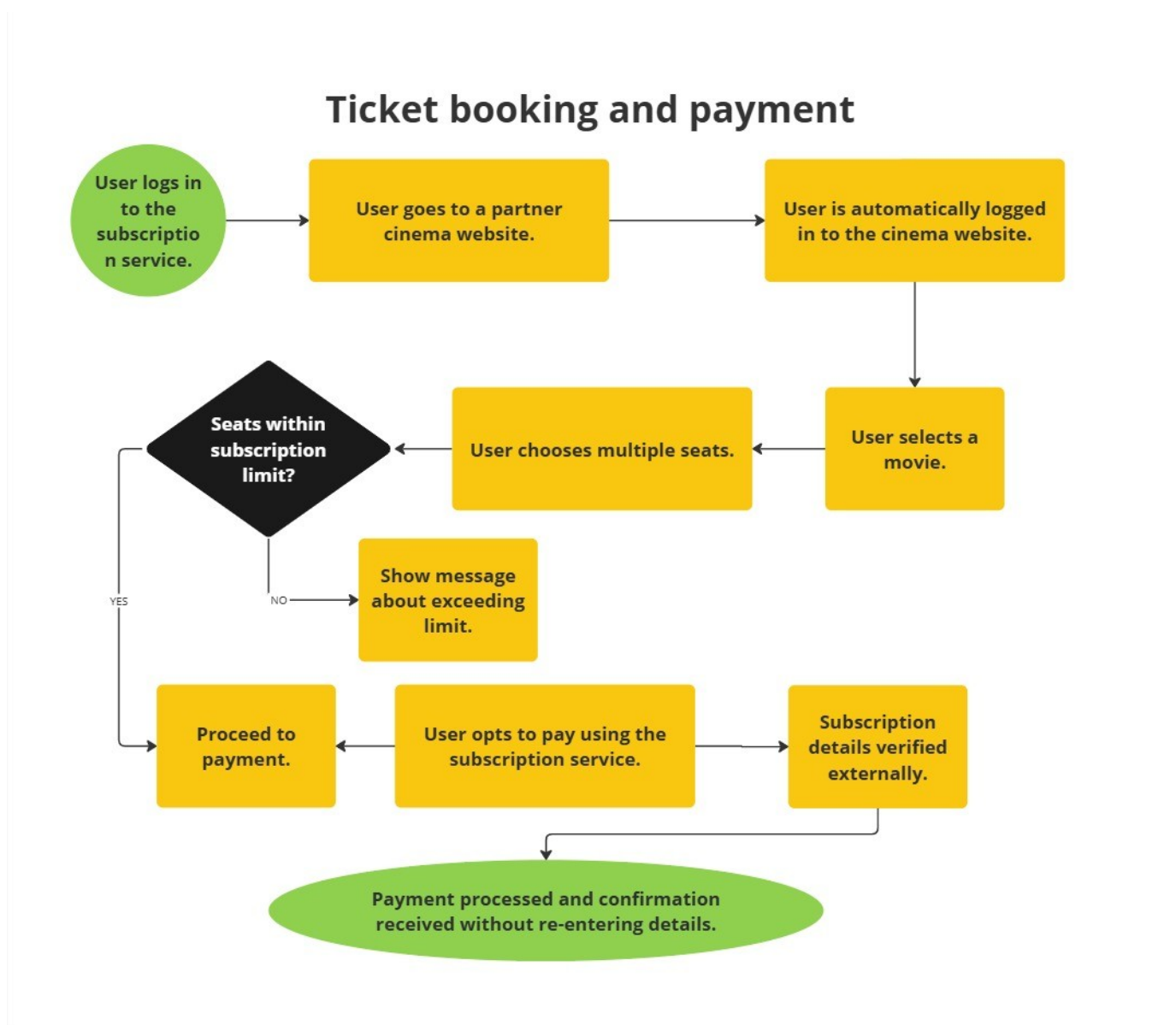
Additional Notes:

- Discuss different subscription plans with the client from an MVP perspective, recommending a minimal viable plan initially.
- Ensure seamless integration with the external subscription service for payment processing.

User Story 2: *Check-in for Movie Visit*

Title: Payment for Cinema Ticket using External Subscription Service

As a subscribed customer **I want to** pay for my cinema ticket using my subscription **So that** I can enjoy movies without re-entering my payment details



Acceptance Criteria:

1. **Given** I am logged in to the subscription service
2. **When** I navigate to a partner cinema website
3. **Then** I should be automatically logged in to the cinema website
4. **Given** I select a movie and proceed to checkout
5. **Given** I can select multiple seats
6. **Then** I should see an option to pay using my subscription service
7. **Given** I choose the subscription service as my payment method
8. **When** I confirm my selection
9. **Then** my subscription details should be verified externally
10. **When** I selected more then my subscription limit allow I got a message
11. **And** the payment should be processed without needing to re-enter payment information
12. **And** I should receive a confirmation of my ticket purchase

Additional Notes:

- Choose an SSO Single Sign-On Provider: (e.g., Okta, Auth0, OneLogin).
- Ensure the external subscription service is seamlessly integrated.
- Handle any errors gracefully, such as subscription verification failures.
- Provide clear messages to the customer during each step of the process.

User Story 3: *Manage Subscription*

As a subscribed user **I want to** manage my subscription **So that** I can update payment information or cancel if needed

Acceptance Criteria:

1. Subscription details and usage history.
 - **Given** the user is viewing their subscription details
 - **Then** they can see current subscription details and usage history
2. Update of payment method.
 - **When** the user updates their payment method
 - **Then** the update process is secure and complies with PCI DSS standards

3. Cancellation of a subscription.
 - **When** the user cancels their subscription
 - **Then** the system shows the next billing date and clear cancellation rules
4. A payment failure.
 - **Given** a payment failure occurs
 - **Then** the system retries up to 3 attempts before cancelling the subscription
5. No refund in MVP
 - **Given** the MVP stage
 - **Then** refunds are not applicable, but future iterations will consider a more customer-friendly approach
6. Lock an Account after multiple payments failures
 - **When** multiple payment failures occur
 - **Then** the system locks the account and notifies the user

Corner Cases:

1. **Service Downtime:** Provide fallback options or clear messages if the subscription management service or payment gateway is down.
2. **Failed Payment Alerts:** Notify users promptly of payment failures and provide clear instructions for resolving the issue.
3. **Data Security:** Implement strict data security measures to protect user payment information during updates.

User Story 4: *Cross-Platform Consistency*

As a user I want to have a seamless experience on both iOS and Android **So that** I can use my subscription easily regardless of my device

Acceptance Criteria:

1. Consistent experience on iOS & Android.
 - **Given** the user is using the app on iOS or Android
 - **Then** all features should work identically on both platforms
 - **And** the UI should comply with respective platform guidelines, but core functionality and user experience must remain consistent
2. Same alerts and errors for both platforms.

- **When** errors occur or alerts are triggered
- **Then** error handling and alerts should be optimized for both platforms (e.g., Apple Push Notifications for iOS, Google FCM for Android)

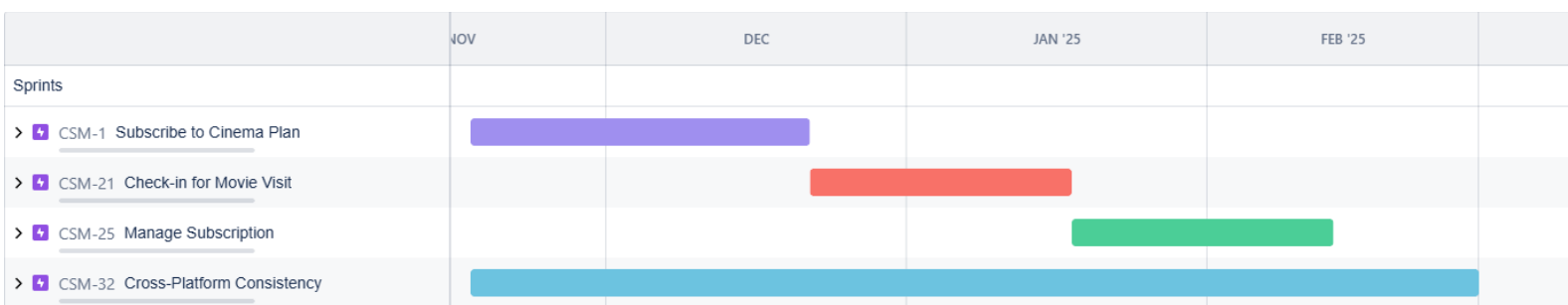
Edge Cases:

1. **Device Switch:** When a user logs in from a new device, their subscriptions, usage history, and settings should sync automatically

Additional Notes:

- Ensure thorough testing on both platforms to guarantee feature parity.
- Follow platform-specific guidelines to enhance user experience while maintaining consistency across devices.

4. Timeline (Scrum-Oriented)



Sprint 1 - Project Discovery & Initial Backlog Setup

- **Duration:** 3 weeks
- **Development Goals:**
 - Project discovery, high-level requirements alignment, initial backlog for subscription and booking flows.
 - Development of user stories for subscribing to a cinema plan and seat booking flow.
- **Customer Review:**
 - **Date:** Mid of Sprint 1 (review of functionality user stories and timeline)
 - **Date:** End of Sprint for feedback integration from first meeting
 - **Focus:** Present MVP backlog, confirm scope and priorities, align on security standards and payment expectations.

Sprint 2 - MVP Subscription Flow Development

- **Duration:** 3 weeks
 - **Development Goals:**
 - Complete subscription functionality: choosing a plan, handling payments, implementing SSL/TLS-secured data transfers.
 - Develop edge cases, e.g., payment failures and partial payments.
 - **Customer Review:**
 - **Date:** End of Sprint 2
 - **Focus:** Validate subscription flow and acceptance criteria, refine edge cases for reliability in payment and service.
-

Sprint 3 - Subscription Management and Payment Integration

- **Duration:** 3 weeks
 - **Development Goals:**
 - Implement subscription management features, including recurring billing, secure updates, and subscription cancellations.
 - Introduce retry options for failed payments.
 - **Customer Review:**
 - **Date:** Mid-Sprint 3 (week 1)
 - **Focus:** Present subscription management features, progress on payment and security compliance; collect feedback.
-

Sprint 4 - Check-In Process and Cross-Platform Development

- **Duration:** 3 weeks
 - **Development Goals:**
 - Build check-in process with SSO integration, booking flow consistency across platforms (iOS and Android).
 - Ensure error handling, consistent alerts, and seamless cross-device experience.
 - **Customer Review:**
 - **Date:** End of Sprint 4
 - **Focus:** Review SSO integration, cross-platform experience, and UI compliance with platform-specific guidelines.
-

Sprint 5 - Full MVP Feature Completion & Refinement

- **Duration:** 3 weeks
- **Development Goals:**
 - Complete all MVP features, refine UX, and incorporate additional security as needed.

- Full testing of booking flows, subscription management, and cross-platform consistency.
- **Customer Review:**
 - **Date:** Mid-Sprint 5 (week 1)
 - **Focus:** Review complete MVP functionality, customer feedback adjustments, and edge case handling.

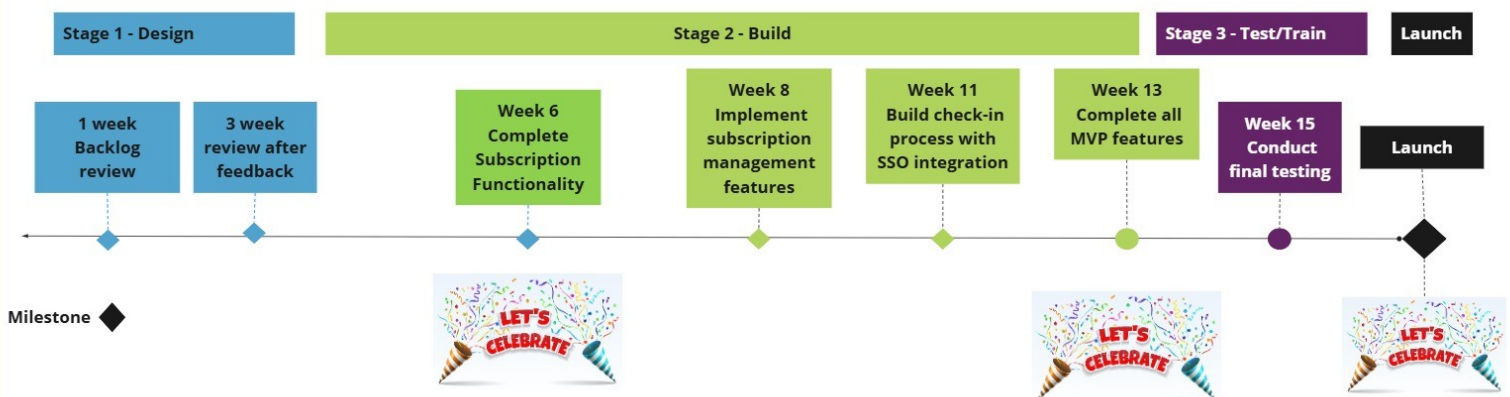
Sprint 6 - Final Adjustments and Pre-Launch Sign-Off

- **Duration:** 3 weeks
 - **Development Goals:**
 - Conduct final testing (regression, security) across platforms, finalize adjustments, prepare for project release.
 - **Customer Review:**
 - **Date:** End of Sprint 6
 - **Focus:** Final project sign-off with customer, ensuring all acceptance criteria and security standards are fully met.
-

Total Project Duration: 18 weeks (6 sprints, each 3 weeks)

This timeline supports efficient sprint cycles while incorporating essential customer feedback, aligning with agile principles for adaptive planning and iterative progress.

Step 5: Reviews Timeline



5. Security and Payment Enhancements

1. Security Requirements (PCI DSS):

- Ensure all payment methods comply with the PCI DSS (Payment Card Industry Data Security Standard).
- Encrypt sensitive user data (payment methods, personal details) both at rest and in transit.
- Implement multi-factor authentication (MFA) for users managing their subscriptions.
- Regularly audit the system to check for vulnerabilities and ensure compliance.

2. Payment Methods:

- Support cards pay with payment gateways like PayPal or Stripe.
- Handle failed payments gracefully with automated retry mechanisms and user notifications.
- Implement fraud detection mechanisms to monitor for unusual activity.

3. Fintech-Related Aspects:

- Detailed logging for all payment-related actions.
- Ensure seamless integration with external APIs from cinemas to process payments securely.
- Maintain a comprehensive audit trail for financial transactions to comply with legal and regulatory standards.