

## 신규출시 게임예약

1. 사용자는 결제 방식을 선택하고 결제

2. 결제 금액은 담당자가 등록한 할인금액 적용

3. 결제는 해당국사 지원신용카드와 간편 결제 서비스

4. 포인트 사용가능, 포인트 외 금액은 신용카드나 간편결제 서비스에 청구

5. 결제를 완료하면 예약완료

6. 결제가 완료되면 사용자에게 포인트 적립되고 메일발송

7. 구매 목록에 신규 게임이 나타나고, 환불요청 버튼 활성화

위 내용으로 예약결제에 필요한 api 기능을 동기/비동기방식으로 정의하고 실패 시 필요한 보상 트랜잭션 정리해주세요

답안 표시

## 신규 출시 게임 예약 시스템 API 기능 정의 및 동기/비동기 방식과 실패 시 보상 트랜잭션

### 1. API 기능 정의 및 동기/비동기 방식

#### 1.1. 게임 목록 조회 API (동기)

- 목적: 사용자가 예약 가능한 게임 목록을 조회합니다.
- 요청: 사용자 정보 (ID, 토큰)
- 응답: 게임 목록 (게임 ID, 게임명, 가격, 할인율, 할인 기간 등)
- 동기 방식 이유: 사용자는 즉시 게임 목록을 확인해야 하므로 동기 방식이 적합합니다.

#### 1.2. 게임 상세 정보 조회 API (동기)

- 목적: 특정 게임의 상세 정보를 조회합니다.
- 요청: 게임 ID
- 응답: 게임 상세 정보 (게임 설명, 이미지, 시스템 요구 사항 등)
- 동기 방식 이유: 사용자는 게임 상세 정보를 즉시 확인해야 하므로 동기 방식이 적합합니다.

### 1.3. 결제 정보 입력 API (동기/비동기 혼합)

- **목적:** 사용자가 결제 정보를 입력하고 결제를 요청합니다.
- **요청:** 사용자 정보, 게임 ID, 결제 방식, 결제 정보, 포인트 사용 여부
- **응답:** 결제 요청 접수 (동기)
- **비동기 처리:** 실제 결제는 백그라운드에서 비동기적으로 처리하며, 결과는 별도의 API를 통해 조회하거나 웹훅으로 알림을 받습니다.
- **동기/비동기 혼합 이유:** 사용자는 즉시 결제 요청을 보내고, 결제 결과는 나중에 확인하거나 알림을 받는 방식이 일반적입니다.

### 1.4. 결제 상태 조회 API (동기)

- **목적:** 사용자가 결제 상태를 조회합니다.
- **요청:** 결제 ID
- **응답:** 결제 상태 (결제 완료, 결제 실패, 결제 대기 등)
- **동기 방식 이유:** 사용자는 즉시 결제 상태를 확인해야 하므로 동기 방식이 적합합니다.

### 1.5. 환불 요청 API (동기/비동기 혼합)

- **목적:** 사용자가 결제를 취소하고 환불을 요청합니다.
- **요청:** 결제 ID
- **응답:** 환불 요청 접수 (동기)
- **비동기 처리:** 실제 환불은 백그라운드에서 비동기적으로 처리하며, 결과는 별도의 API를 통해 조회하거나 웹훅으로 알림을 받습니다.
- **동기/비동기 혼합 이유:** 사용자는 즉시 환불을 요청하고, 환불 결과는 나중에 확인하거나 알림을 받는 방식이 일반적입니다.

## 2. 실패 시 보상 트랜잭션

- **결제 실패:**
  - 사용자에게 결제 실패 사유를 알림
  - 예약 정보 삭제
  - 사용된 포인트 복원
- **환불 실패:**

- 사용자에게 환불 실패 사유를 알림
- 관리자에게 알림
- 재시도 메커니즘 구축 (예: 일정 시간 후 재시도)

- **시스템 오류:**

- 로그 기록
- 관리자에게 알림
- 재시도 메커니즘 구축 (예: 일정 시간 후 재시도)
- **데이터 일관성 유지:**
  - 분산 트랜잭션 또는 이벤트 소싱을 활용하여 데이터 일관성을 유지합니다.
  - 예를 들어, 결제 성공 시 예약 정보와 포인트 정보를 함께 업데이트하는 트랜잭션을 수행하고, 실패 시 롤백합니다.

### 3. 추가 고려 사항

- **비동기 처리:**

- 메시지 큐 (RabbitMQ, Kafka 등)를 활용하여 비동기 작업을 처리합니다.
- 작업 결과는 데이터베이스에 저장하거나 웹훅을 통해 알립니다.

- **데이터베이스:**

- 결제 정보, 예약 정보, 포인트 정보를 효율적으로 관리하기 위해 적절한 데이터베이스를 선택합니다.
- 데이터베이스 트랜잭션을 활용하여 데이터 일관성을 유지합니다.

- **보안:**

- 결제 정보 암호화, SQL 인젝션 방지, XSS 방지, CSRF 방지 등 보안 조치를 철저히 수행합니다.

- **성능:**

- 캐싱, 분산 시스템, 비동기 처리 등을 활용하여 시스템 성능을 향상시킵니다.

- **확장성:**

- 마이크로서비스 아키텍처를 도입하여 시스템을 확장 가능하게 만듭니다.

- **유연성:**

- 새로운 결제 수단, 할인 정책 등에 유연하게 대응할 수 있도록 시스템을 설계합니다.

#### 4. 결론

신규 출시 게임 예약 시스템은 사용자 편의성, 보안, 성능을 모두 고려하여 설계해야 합니다. 위에서 제시된 API 기능 정의 및 동기/비동기 방식을 참고하여 시스템을 구축하고, 지속적인 모니터링과 개선을 통해 안정적인 서비스를 제공할 수 있습니다.

#### 핵심 기능 구현 시 고려해야 할 사항:

- **결제 연동:** 다양한 PG사와의 연동을 위한 SDK 또는 API를 활용해야 합니다.
- **포인트 시스템:** 포인트 적립, 사용, 만료 등에 대한 로직을 구현해야 합니다.
- **오류 처리:** 다양한 오류 상황에 대한 예외 처리를 구현해야 합니다.
- **로그:** 시스템 동작 로그를 기록하여 문제 발생 시 분석에 활용해야 합니다.
- **보안:** 위에서 언급한 보안 사항들을 철저히 준수해야 합니다.

궁금한 점이 있다면 언제든지 질문해주세요.

#### 신규출시 게임예약 선물하기

1. 선물 받은 사람이 선물거절을 하면 보낸 사람에게 바로 환불처리

2. 친구가 30일 동안 수락하지 않으면 자도 환불처리

3. 선물한 내용은 리포팅 시스템에서 조회 가능

위 내용으로 환불처리에 필요한 api 기능을 동기/비동기방식으로 정의하고 실패 시 필요한 보상 트랜잭션 정리해주세요

답안 표시

#### 신규 출시 게임 예약 선물하기 환불 처리 API 및 보상 트랜잭션

##### 1. API 기능 정의 및 동기/비동기 방식

### 1.1. 선물하기 API (동기/비동기 혼합)

- **목적:** 사용자가 다른 사용자에게 게임을 선물합니다.
- **요청:** 선물할 게임 ID, 받는 사람의 ID, 메시지 (선택 사항)
- **응답:** 선물 요청 접수 (동기)
- **비동기 처리:** 실제 선물 처리 (결제, 알림 등)는 백그라운드에서 비동기적으로 처리하며, 결과는 별도의 API를 통해 조회하거나 웹훅으로 알림을 받습니다.
- **동기/비동기 혼합 이유:** 사용자는 즉시 선물 요청을 보내고, 선물 처리 결과는 나중에 확인하거나 알림을 받는 방식이 일반적입니다.

### 1.2. 선물 거절 API (동기)

- **목적:** 선물을 받은 사용자가 선물을 거절합니다.
- **요청:** 선물 ID
- **응답:** 거절 결과 (성공/실패)
- **동기 방식 이유:** 사용자는 즉시 거절하고 환불을 원하므로 동기 방식이 적합합니다.

### 1.3. 선물 상태 조회 API (동기)

- **목적:** 선물 상태를 조회합니다. (선물 중, 거절, 환불 완료 등)
- **요청:** 선물 ID
- **응답:** 선물 상태

### 1.4. 환불 처리 API (비동기)

- **목적:** 선물 거절 등으로 인해 환불을 처리합니다.
- **요청:** 선물 ID
- **응답:** 환불 요청 접수
- **비동기 처리:** 실제 환불은 백그라운드에서 비동기적으로 처리하며, 결과는 별도의 API를 통해 조회하거나 웹훅으로 알림을 받습니다.

## 2. 실패 시 보상 트랜잭션

- **선물 거절 시:**
  - 선물 상태를 "거절"로 업데이트

- 환불 처리 API 호출
- **환불 실패:**
  - 사용자에게 환불 실패 사유를 알림
  - 관리자에게 알림
  - 재시도 메커니즘 구축 (예: 일정 시간 후 재시도)
- **시스템 오류:**
  - 로그 기록
  - 관리자에게 알림
  - 재시도 메커니즘 구축 (예: 일정 시간 후 재시도)
  - **데이터 일관성 유지:**
    - 분산 트랜잭션 또는 이벤트 소싱을 활용하여 데이터 일관성을 유지합니다.
    - 예를 들어, 선물 거절 시 선물 상태와 환불 상태를 함께 업데이트하는 트랜잭션을 수행하고, 실패 시 롤백합니다.

### 3. 추가 고려 사항

- **시간 제한:** 선물 수락 기간 설정 (예: 30일)
- **선물 취소:** 선물 취소 기능 구현 (시간 제한 내)
- **선물 내역 조회:** 사용자가 보낸/받은 선물 내역을 조회할 수 있도록 기능 구현
- **보안:** 선물 정보 보안, 개인정보 보호 등
- **성능:** 높은 동시 접속 처리를 위한 시스템 설계

### 4. 시스템 다이어그램 (예시)

[이미지: 신규 출시 게임 예약 선물하기 시스템 다이어그램]

#### 다이어그램 설명:

- 사용자가 선물을 보내면 백엔드 서버에서 선물 정보를 저장하고, 비동기적으로 결제를 처리합니다.
- 선물을 받은 사용자가 거절하면 백엔드 서버에서 환불 처리를 요청합니다.
- 환불 처리 결과는 사용자에게 알려집니다.

## 5. 결론

신규 출시 게임 예약 선물하기 기능은 복잡한 비즈니스 로직과 다양한 예외 상황을 고려해야 합니다. 위에서 제시된 API 기능 정의 및 동기/비동기 방식을 참고하여 시스템을 구축하고, 지속적인 모니터링과 개선을 통해 안정적인 서비스를 제공할 수 있습니다.

### 핵심 기능 구현 시 고려해야 할 사항:

- **결제 연동:** 다양한 PG사와의 연동을 위한 SDK 또는 API를 활용해야 합니다.
- **포인트 시스템:** 선물과 관련된 포인트 적립/차감 로직을 구현해야 합니다.
- **알림:** 선물 상태 변경 시 사용자에게 알림을 보내야 합니다.
- **로그:** 시스템 동작 로그를 기록하여 문제 발생 시 분석에 활용해야 합니다.
- **보안:** 선물 정보 보안, 개인정보 보호 등 보안 조치를 철저히 수행해야 합니다.

궁금한 점이 있다면 언제든지 질문해주세요.