

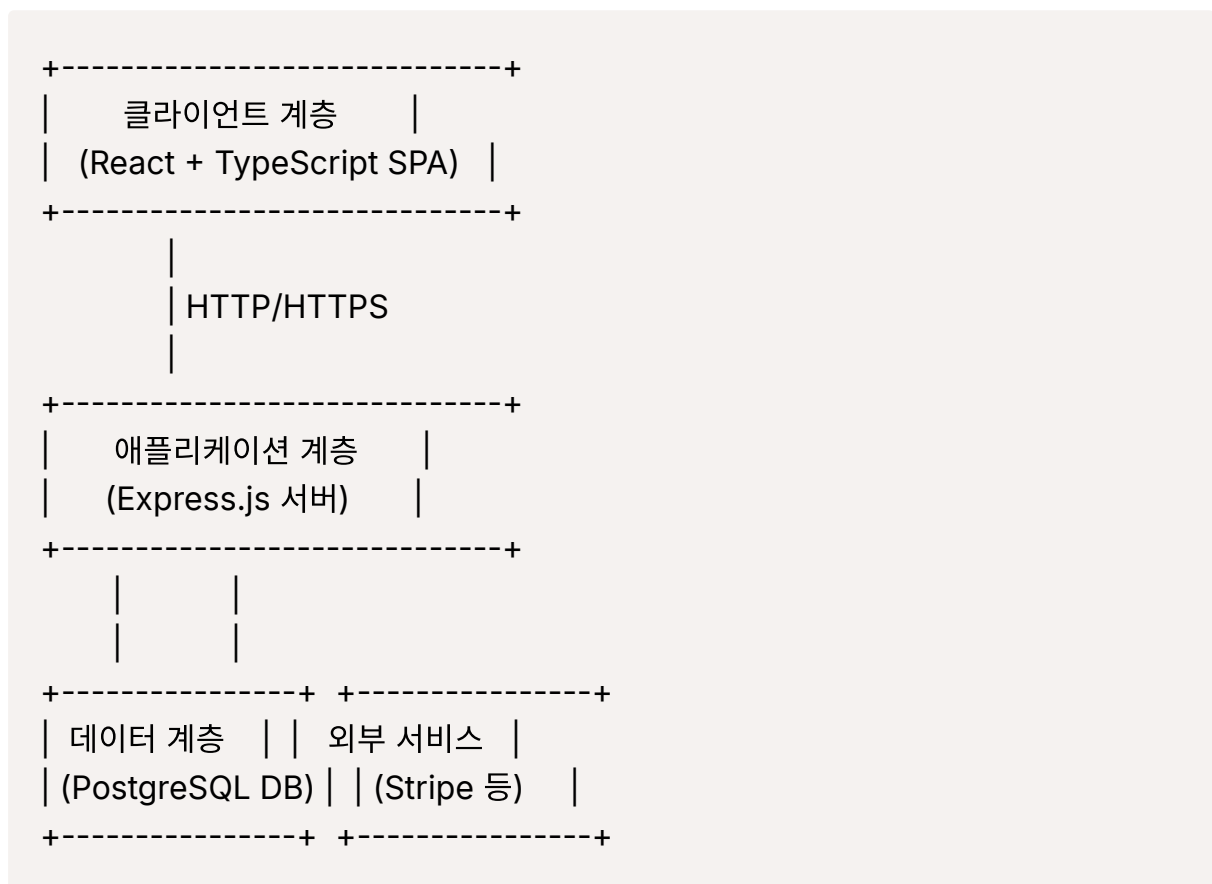
# 시스템 아키텍처 계획서

## 중고 마켓플레이스 시스템 아키텍처 명세서

### 1. 개요

본 문서는 한국어 중고 마켓플레이스 웹 애플리케이션의 시스템 아키텍처를 설명한다. 해당 시스템은 사용자가 중고 상품을 등록, 검색, 구매할 수 있는 온라인 플랫폼으로, 실시간 채팅, 결제 시스템, 사용자 인증 등 다양한 기능을 포함한다.

### 2. 시스템 구성도



### 3. 아키텍처 설계 원칙

- **모듈성:** 독립적인 개발, 테스트, 배포가 가능한 구조
- **확장성:** 사용자 및 데이터 증가에 유연하게 대응
- **보안성:** 사용자 데이터 보호 및 안전한 결제 처리
- **유지보수성:** 명확한 구조와 문서화를 통한 효율적 유지보수

- **사용자 경험:** 빠른 응답과 직관적 UI 제공

## 4. 필요한 기능

- 사람들이 플랫폼에 가입할 수 있어야 함
- 상품들을 올리고 볼 수 있어야 함.
- 플랫폼 사용자들끼리 소통이 가능해야함.
- 악성 유저나 상품을 차단 해야 함.
- 유저들 간의 송금이 가능해야함
- 상품의 검색할 수 있어야 함.
- 관리자가 플랫폼의 모든 요소를 관리할 수 있어야 함.

### 세부 목적

#### 1. 사용자 계정 관리

- 사용자는 플랫폼에 가입하고 로그인할 수 있다.
- 사용자는 자신의 프로필 정보를 관리할 수 있다.
- 사용자는 회원 탈퇴를 할 수 있다.
- 비밀번호는 안전하게 암호화되어 저장된다.
- 사용자 인증은 세션 기반으로 처리한다.
- 소셜 로그인 한다.

#### 2. 상품 관리

- 사용자는 판매할 상품을 등록할 수 있다.
- 사용자는 상품 정보(제목, 설명, 가격, 카테고리, 이미지)를 입력할 수 있다.
- 사용자는 자신이 등록한 상품 정보를 수정하거나 삭제할 수 있다.
- 상품 상태(판매중, 예약중, 판매완료)를 관리할 수 있다.
- 사용자는 모든 등록된 상품을 열람할 수 있다.
- 최근 등록된 상품을 메인 페이지에서 확인할 수 있다.
- 위치 기반 서비스 - 삭제
- 평점 관리 시스템 - API 추가

### 3. 실시간 소통 기능

- 사용자는 상품 판매자와 실시간 채팅을 할 수 있다.
- 채팅 메시지는 폴링 방식으로 실시간으로 업데이트된다.
- 사용자는 자신의 모든 채팅방 목록을 볼 수 있다.
- 읽지 않은 메시지 수가 표시된다.
- 로그인 시 새로운 채팅 메시지가 있으면 알림을 받는다.
- 다국어 지원한다. → 삭제

### 4. 차단 및 신고 기능

- 사용자는 다른 사용자를 차단할 수 있다.
- 사용자는 부적절한 상품이나 사용자를 신고할 수 있다.
- 차단된 사용자와는 채팅을 할 수 없다.
- 차단 목록을 관리할 수 있다.

### 5. 결제 시스템

- Stripe를 활용한 안전한 결제 시스템이 구현 된다.
- 사용자는 신용카드로 상품 결제를 할 수 있다.
- 결제 과정은 단계별로 안내된다.
- 테스트 모드에서는 실제 결제가 발생하지 않는다.

### 6. 검색 및 필터링

- 사용자는 상품명으로 검색할 수 있다.
- 카테고리별로 상품을 필터링할 수 있다.
- 지역별로 상품을 필터링할 수 있다.
- 채팅방을 검색할 수 있다.

### 7. 관심 상품 기능

- 사용자는 관심 있는 상품을 찜할 수 있다.
- 찜한 상품 목록을 확인할 수 있다.
- 각 상품별 찜 개수를 확인할 수 있다.

### 8. 관리자 기능

- 관리자는 모든 사용자 목록을 확인할 수 있다.
- 관리자는 사용자 계정을 관리할 수 있다 (조회, 수정, 삭제).
- 관리자는 신고된 내용을 확인하고 처리할 수 있다.
- 관리자는 다른 사용자를 관리자로 지정할 수 있다.

## 9. UI/UX 기능

- 반응형 디자인으로 모바일, 태블릿, 데스크톱 환경에 최적화되어 있다.
- 다크 모드를 지원한다.
- 사용자 친화적인 알림 시스템이 있다.
- 한국어 인터페이스를 제공한다.

## 10. 보안 기능

- XSS 공격 방어 기능이 적용되어 있다.
- CSRF 방어 기능이 있다.
- 로그인 시도 제한(Rate Limiting)이 적용되어 있다.
- 세션 보안이 강화되어 있다.
- 안전한 쿠키 설정이 적용되어 있다.
- 보안 헤더(Helmet)가 추가되어 있다.

## 5. 클라이언트 계층

- 기술 스택: React 18, TypeScript 5, React Query, Wouter, TailwindCSS, Stripe Elements
- 주요 컴포넌트: 인증, 상품, 채팅, 결제, 관리자 대시보드
- 아키텍처 특징: 컴포넌트 기반, 훅 기반 상태 관리, 반응형 디자인, 다크 모드 지원

## 6. 애플리케이션 계층

- 기술 스택: Node.js 18, Express.js 4, TypeScript, RESTful API
- 주요 모듈: 인증, 상품, 채팅, 결제, 관리자, 보안
- 서버 아키텍처: 라우팅-컨트롤러-서비스-DAO 구조, 미들웨어 처리

## 7. 데이터 계층

- **DB 종류:** PostgreSQL
- **ORM:** Drizzle ORM
- **주요 테이블:** users, products, favorites, chat\_rooms, chat\_messages, comments, reports, blocks
- **데이터 모델:** 사용자, 상품, 거래, 채팅, 신고 관련 엔티티 정의

## 8. 외부 서비스 통합

- **결제:** Stripe API 기반 카드 결제 처리
- **이미지:** 서버 저장 및 최적화 처리 (크기 조절, 포맷 변환 등)

## 9. 보안 아키텍처

- **인증/권한:** 세션 기반 인증, RBAC, bcrypt 해싱
- **데이터 보호:** HTTPS, 입력 검증, XSS/CSRF 방지, 요청 제한, 보안 헤더 설정

## 10. 성능 최적화

- **프론트엔드:** 코드 분할, React Query 캐싱, 번들 최적화
- **백엔드:** 쿼리 튜닝, 메모리 캐싱, 요청 병합

## 11. 배포 및 인프라

- **개발/테스트:** Docker, Jest, GitHub Actions
- **운영:** Replit, Express 서버 설정, 로깅 및 오류 추적

## 12. 확장성 계획

- **기술 확장:** DB 스케일링, 서버 수평 확장
- **기능 확장:** 소셜 로그인, 모바일 앱, AI 추천, 에스스로 결제

## 13. 아키텍처 제약사항 및 고려사항

- **기술적 제약:** 자원 제한, 네트워크 지연, 브라우저 호환성
- **비기능 요구사항:** 응답 시간 < 2초, 가용성 99.9%, OWASP 대응, 5,000명 이상 동시 접속 지원

## 14. 아키텍처 의사결정 기록

- **React Query 선택:** 서버 상태 관리 최적화

- **RESTful 채팅 선택:** 구현 단순화와 유지보수 고려
- **세션 인증 선택:** 보안성과 서버 제어 중심 설계

## 15. 참조 및 관련 문서

- API 명세서
- DB 스키마 설계서
- UI/UX 가이드라인
- 보안 정책 문서
- 테스트 전략 및 계획