이모티콘 기능 통합 설계 문서

1. 기능 개요

사용자는 이모티콘을 등록, 구매, 사용 및 관리할 수 있다. 판매자와 일반 사용자로 역할이 구분되며, 이모티콘은 GIF를 포함한 이미지로 등록 가능하다. 포인트 기반 구매 방식과 찜, 공유 기능이 포함되며, 인기/최신 정렬 및 작가의 다른 작품 보기 기능도 제공된다.

2. 프론트 페이지 구성 및 흐름

1) 이모티콘 등록 페이지 (판매자 전용)  
- 이름, 설명, 가격, 대표 이미지 포함 업로드  
- 등록 시 POST /api/emojis/register 요청  
  
2) 이모티콘 샵 (모든 사용자)  
- 전체 목록 확인, 찜하기, 구매, 링크 복사 가능  
- 홈 화면에는 인기순/최신순 정렬 제공  
  
3) 내 이모티콘 보관함  
- 구매한 이모지만 표시  
  
4) 채팅방  
- 입력창 옆 이모티콘 버튼으로 구매한 이모지 전송 가능  
  
5) 이모티콘 상세 페이지  
- 대표 이미지, 이름, 설명, 가격 표시  
- 하단에 해당 작가의 다른 이모티콘 리스트 출력  
- 찜하기/공유하기 버튼 포함

3. 상세 기능 요구사항 및 시나리오

● 등록: 판매자가 등록 폼 입력 후 POST /api/emojis/register 호출  
● 구매: 사용자 POST /api/emojis/buy 호출, 포인트 차감 후 구매 처리  
● 사용: 구매한 이모지만 채팅 입력창에서 사용 가능  
● GIF 지원: imageUrl을 <img> 태그로 렌더링  
● 찜: 버튼 클릭 시 EmojiBookmark 테이블에 저장  
● 공유: 상세 페이지 링크 복사 기능 제공

4. 요청 및 응답 DTO 예시

이모티콘 등록 요청:

{  
 "name": "귀여운곰돌이",  
 "imageUrl": "https://image.com/bear.gif",  
 "description": "귀여운 곰돌이 이모티콘입니다",  
 "price": 2000,  
 "mainImageUrl": "https://image.com/bear\_main.gif"  
}

이모티콘 구매 요청:

{  
 "emojiId": 5,  
 "userId": 3  
}

공통 응답 예시:

{  
 "status": 200,  
 "message": "Success",  
 "data": {  
 "emojiId": 5,  
 "name": "곰돌이",  
 "imageUrl": "https://image.com/bear.gif"  
 }  
}

5. 예외 상황

- 필수 값 누락 → 400 Bad Request  
- 중복된 이름 등록 → 409 Conflict  
- 존재하지 않는 이모티콘 구매 요청 → 404 Not Found  
- 이미 구매한 이모티콘 재구매 → 409 Conflict  
- 포인트 부족 → 400 Bad Request  
- 비로그인 사용자 요청 → 401 Unauthorized

6. API 엔드포인트 목록

- POST /api/emojis/register  
- POST /api/emojis/buy  
- GET /api/emojis  
- GET /api/emojis/popular  
- GET /api/emojis/latest  
- GET /api/emojis/my  
- GET /api/emojis/{id}

7. ERD 관계 설명

User  
 ├── 1:N ──> Emoji (판매자: 이모티콘 등록)  
 ├── 1:N ──> EmojiPurchase (구매 기록)  
 └── 1:N ──> EmojiBookmark (찜 기록)  
  
Emoji  
 ├── 1:N ──> EmojiPurchase (이모티콘 구매됨)  
 └── 1:N ──> EmojiBookmark (이모티콘 찜됨)

요약:  
- User 1:N Emoji  
- User 1:N EmojiPurchase  
- User 1:N EmojiBookmark  
- Emoji 1:N EmojiPurchase  
- Emoji 1:N EmojiBookmark

8. DB 테이블 설계

User:  
- id (Long)  
- username (String)  
- role (USER/SELLER)  
- point (기본 50000)  
  
Emoji:  
- id (Long)  
- name (String)  
- imageUrl (String)  
- description (String)  
- mainImageUrl (String)  
- sellerId (Long, not null, FK)  
- price (Integer)  
- createdAt (LocalDateTime)  
- isAnimated (Boolean)  
  
EmojiPurchase:  
- id (Long)  
- userId (FK)  
- emojiId (FK)  
- purchasedAt (LocalDateTime)  
  
EmojiBookmark:  
- id (Long)  
- userId (FK)  
- emojiId (FK)

9. dbdiagram.io 코드 예시

Table user {  
 id int [pk]  
 username varchar  
 point int  
 role varchar  
}  
  
Table emoji {  
 id int [pk]  
 name varchar  
 imageUrl varchar  
 description varchar  
 mainImageUrl varchar  
 sellerId int [not null, ref: > user.id]  
 price int  
 createdAt datetime  
 isAnimated boolean  
}  
  
Table emoji\_purchase {  
 id int [pk]  
 userId int [ref: > user.id]  
 emojiId int [ref: > emoji.id]  
 purchasedAt datetime  
}  
  
Table emoji\_bookmark {  
 id int [pk]  
 userId int [ref: > user.id]  
 emojiId int [ref: > emoji.id]  
}

10. ERD 다이어그램 이미지

