☒ 캠퍼스 마켓 (Uni-Market) 프로젝트 기획안

팀 데브루키 (DevRookies)

2024년 2월 15일

차 례

1 1. 프로젝트 개요 (Overview)

1.1 1.1 프로젝트 소개

한 줄 요약: 우리 학교 학생들끼리 믿고 거래하는 중고 전공 서적 & 자취 물품 거래 플랫폼

대학생들의 경우 매 학기 비싼 전공 서적을 구매해야 하는 부담이 있지만, 기존 중고 거래 플랫폼(당근마켓, 중고나라 등)은 '학교 인증' 절차가 없어 직거래 시 신뢰도 문제가 발생하거나, 캠퍼스 내에서의 거래만을 필터링하기 어려운 불편함이 있었습니다. **Uni-Market**은 학교 이메일 인증을 통해 **"진짜 우리 학교 학생"**들만 거래할 수 있는 폐쇄형 커뮤니티 마켓입니다.

> ☒ **기획 의도**
> "왜 전공책은 한 학기만 쓰고 버려질까?"
> "학교 앞에서 직거래할 때 상대방이 같은 학교 학생인지 알 수 없어 불안했던 경험을 해결하자!"

1.2 1.2 팀원 구성 (Team DevRookies)

- **김팀장 (PM/Backend):** 프로젝트 총괄, DB 설계, API 명세서 작성, 회원가입/인증 구현
- **이프론 (Frontend):** React 기반 UI/UX 구현, 상태 관리, 상품 리스트 페이지
- **박서버 (Backend):** 상품 CRUD, 검색 필터링, 채팅 소켓 구현
- **최디자 (Design/Frontend):** Figma 와이어프레임, 마이페이지 및 채팅 UI 구현

1.3 1.3 개발 기간

- **기간:** 2024.02.15 ~ 2024.03.15 (총 4주)
- **목표:** MVP(Minimum Viable Product) 완성 및 배포

2 2. 시장 분석 및 타겟 유저

2.1 2.1 페르소나 (Persona)

이름/나이	김전공 (22세, 대학교 2학년)
성격/특징	• 학점 관리를 열심히 하지만 용돈이 부족함. • 낯선 사람과의 거래를 두려워함. • 학교 도서관이나 학생회관 근처에서 주로 활동함.
Needs (니즈)	• 정가 4만 원짜리 전공책을 싸게 사고 싶음. • 택배보다는 교내 직거래를 선호함 (배송비 절약). • 상대방이 같은 학교 학생이라는 확신이 필요함.
Pain Points	• 학교 커뮤니티(에브리타임 등)는 게시판 형태라 검색이 불편함. • 당근마켓은 학교 밖 외부인과 섞여 있어 직거래가 부담스러움.

2.2 2.2 경쟁사 분석

1. **에브리타임 (장터 게시판)**
   - 장점: 접근성이 좋음 (대부분의 대학생 사용).
   - 단점: '쇼핑'에 특화된 UI가 아님 (사진 보기 불편, 결제 기능 없음, 카테고리 필터 미흡).
2. **당근마켓**
   - 장점: 위치 기반으로 거래가 용이함.
   - 단점: '우리 학교 학생'만 필터링하는 기능이 부족함. 외부인 노출 가능성.

3 3. 핵심 기능 (Key Features)

우리는 4주라는 짧은 기간 내에 구현 가능한 **핵심 기능(MVP)**에 집중합니다.

3.1 3.1 사용자 기능

- **회원가입/로그인:** JWT 기반 일반 로그인.
- **학교 인증 (핵심):** 'ac.kr' 도메인 이메일 인증을 통해 '인증 뱃지' 부여. (인증된 사용자만 글쓰기 가능)
- **마이페이지:** 판매 내역, 구매 내역, 관심 목록(찜), 프로필 수정.

3.2 3.2 상품 거래 기능

- **상품 등록:** 사진 업로드(최대 5장), 카테고리 설정(전공서적, 전자기기, 자취용품 등), 가격 설정.
- **상품 검색/필터:** 키워드 검색, 카테고리별 보기, '판매중'/'판매완료' 필터.
- **찜하기:** 관심 있는 상품 저장.

3.3 3.3 커뮤니케이션

- **1:1 채팅:** 상품 상세 페이지에서 '채팅하기' 버튼을 눌러 판매자와 대화.
- **약속 잡기:** 직거래 시간 및 장소(교내 랜드마크) 설정.

4 4. 유저 플로우 (User Flow)

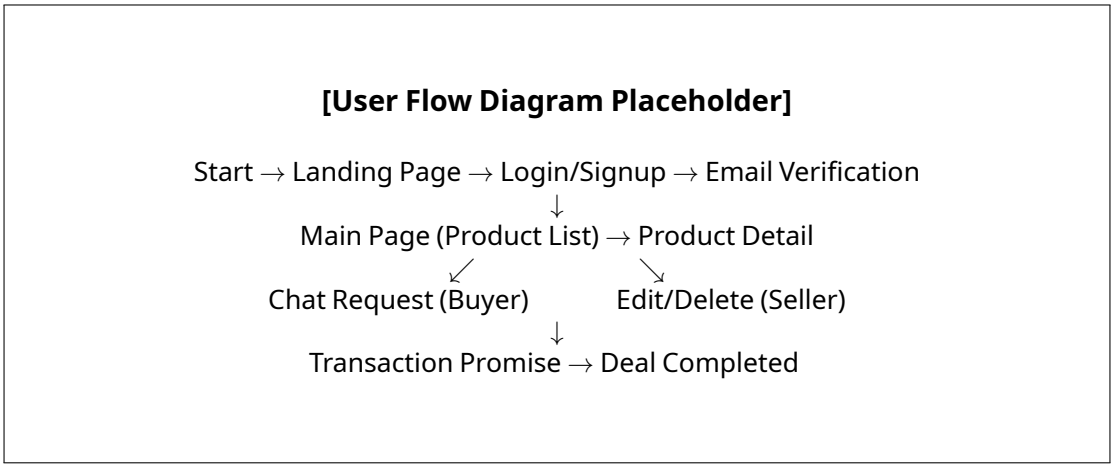

그림 1: Uni-Market 서비스 흐름도

5 5. 데이터베이스 설계 (ERD Draft)

관계형 데이터베이스(MySQL)를 사용하며 주요 엔티티는 다음과 같습니다.

- **Users (사용자):** `user_id`, `email`, `nickname`, `password`, `university_name`, `is_verified`
- **Products (상품):** `product_id`, `seller_id(FK)`, `title`, `price`, `category`, `status(ENUM: SALE, RESERVED, SOLD)`
- **Likes (찜):** `like_id`, `user_id(FK)`, `product_id(FK)`
- **ChatRooms (채팅방):** `room_id`, `product_id(FK)`, `buyer_id(FK)`
- **Messages (메시지):** `msg_id`, `room_id(FK)`, `sender_id(FK)`, `content`, `sent_at`

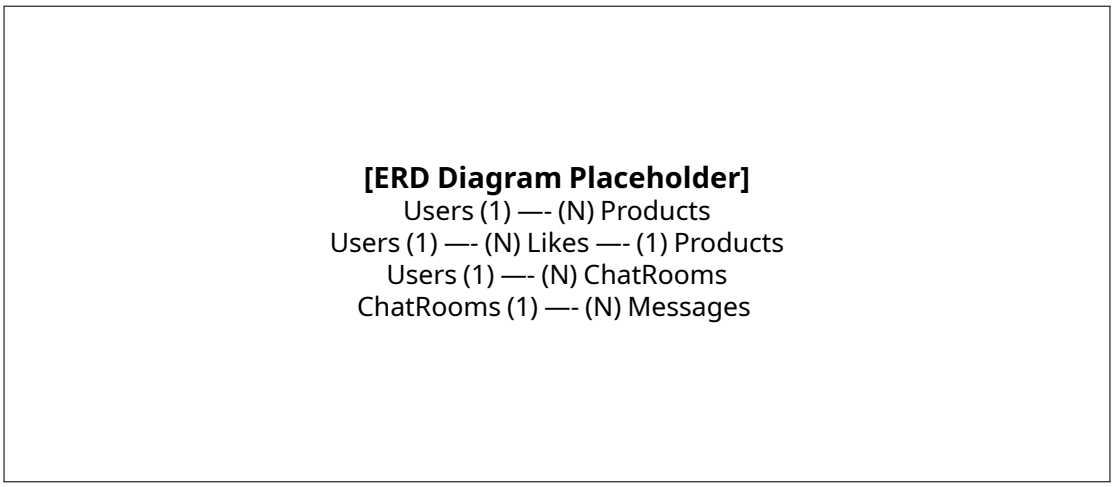

그림 2: 간략화된 ERD 구조

6 6. 화면 설계 (Wireframe Concept)

6.1 6.1 메인 페이지 (Main)

- 상단: 헤더 (로고, 검색창, 로그인/마이페이지).
- 배너: "지금 신학기 전공책을 가장 싸게 구하세요!" 문구.
- 컨텐츠: 최신 등록된 상품 그리드 뷰 (사진, 제목, 가격, 등록시간).

6.2 6.2 상품 상세 페이지 (Detail)

- 좌측: 상품 이미지 슬라이더.
- 우측:
  - 판매자 정보 (닉네임, 학교 인증 뱃지, 매너온도).
  - 상품 정보 (제목, 가격, 카테고리, 상태).
  - 설명 텍스트.
  - 하단 버튼: [찜하기] [채팅하기].

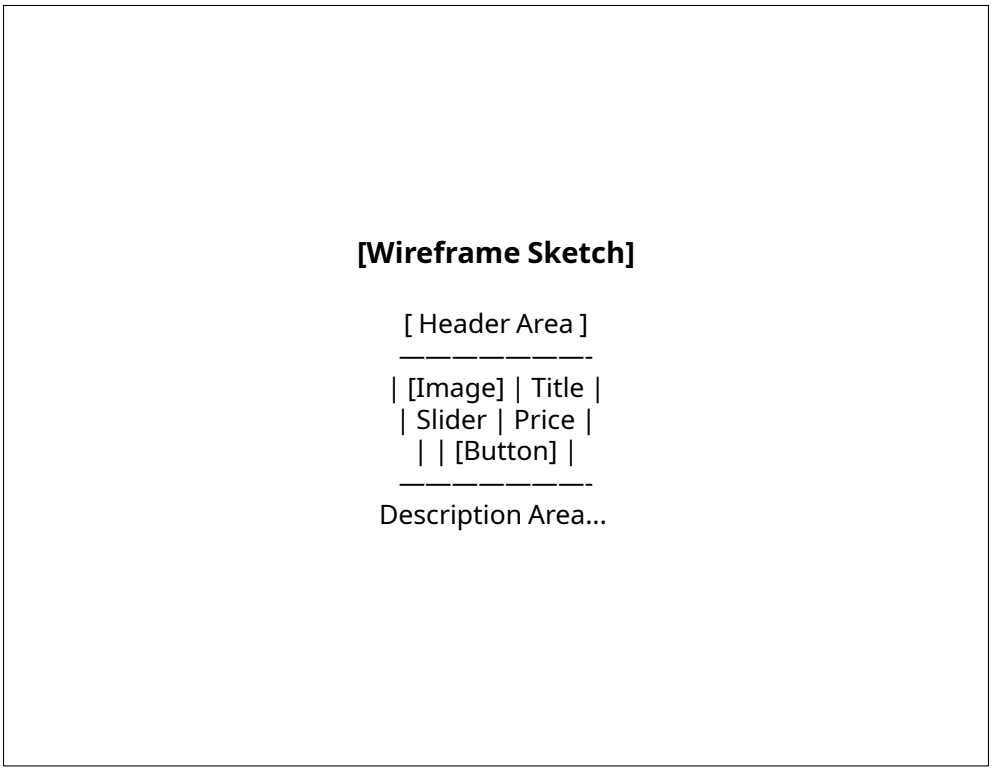

그림 3: 상품 상세 페이지 와이어프레임 초안

7 7. 기술 스택 (Tech Stack)

팀원들의 학습 수준과 취업 시장 수요를 고려하여 선정했습니다.

구분	기술	선정 이유
Frontend	React, TailwindCSS	컴포넌트 재사용성 및 빠른 UI 스타일링 가능.
Backend	Java, Spring Boot	안정적인 서버 구축, 국내 취업 시장 주류 기술.
Database	MySQL	관계형 데이터 관리에 적합하며 레퍼런스가 많음.
Infra	AWS EC2, RDS	클라우드 배포 경험 습득 (프리티어 사용).
Co-op	Git, Notion, Discord	효율적인 형상 관리 및 커뮤니케이션.

8 8. 개발 일정 (WBS)

총 4주 스프린트로 진행하며, 매주 금요일 스크럼을 통해 진행 상황을 공유합니다.

Week 1: 기획 및 설계

완료 아이디어 확정 및 요구사항 정의서 작성.

진행중 Figma UI 디자인 및 와이어프레임 제작.

예정 ERD 설계 및 API 명세서 초안 작성.

Week 2: 기본 세팅 및 회원 기능

- React 프로젝트 초기 세팅 (Router, Axios 등).
- Spring Boot 프로젝트 세팅 및 DB 연동.
- 회원가입(이메일 인증 포함), 로그인 API 구현.

Week 3: 상품 거래 핵심 기능

- 상품 등록(이미지 업로드 AWS S3), 수정, 삭제 구현.
- 메인 페이지 상품 리스트 조회 및 필터링.
- 상품 상세 페이지 UI 연결.

Week 4: 채팅 및 배포/마무리

- WebSocket 기반 1:1 채팅 기능 구현.
- AWS EC2 배포 및 도메인 연결.
- 최종 버그 수정 및 시연 영상 제작.

9 9. 예상되는 어려움 및 해결 방안 (Troubleshooting Plan)

Q. 실시간 채팅은 어떻게 구현할 것인가?
A. 처음에는 단순한 Polling 방식으로 시도해보고, 서버 부하가 우려될 경우 WebSocket(STOMP)을 도입하여 학습해볼 예정입니다.

Q. 이미지 저장소 관리?
A. DB에 이미지를 직접 저장하면 용량 문제가 발생하므로, AWS S3를 활용하여 이미지 URL만 DB에 저장하는 방식을 사용합니다.

— End of Document —
본 문서는 팀 프로젝트 기획용 초안입니다.