

# 경매중 (경제적이고 매력적인 중고거래 플랫폼)

판매자를 위한 흥정, 구매자를 위한 거래

## Table of Content

1. 서비스 개요
2. 기술 스택
3. 한눈에 보는 시스템 아키텍처
4. 팀원 소개
5. Jira Convention
6. Git Flow Convention
7. Commit Message Convention
8. Backend Convention
9. Frontend Convention
10. Wire Frames
11. 기능 정의서
12. ERD
13. Sequence Diagram
14. References

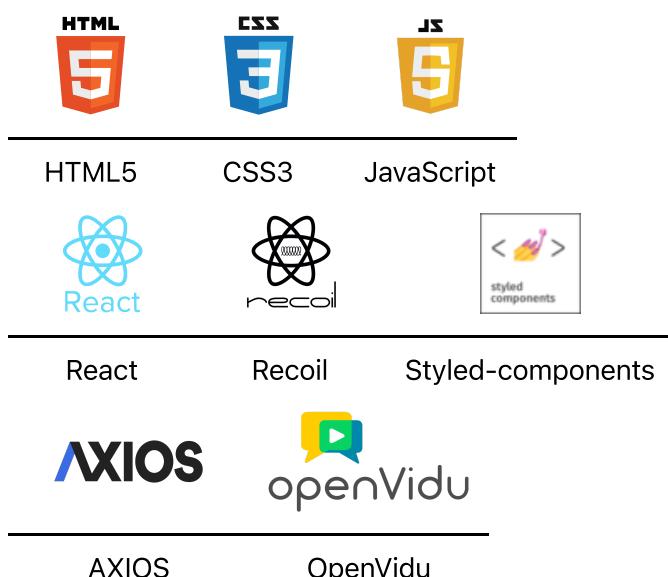
## 1. 서비스 개요

저희 서비스가 추구하는 목표는 경제적이고 매력적인 중고거래 플랫폼입니다.

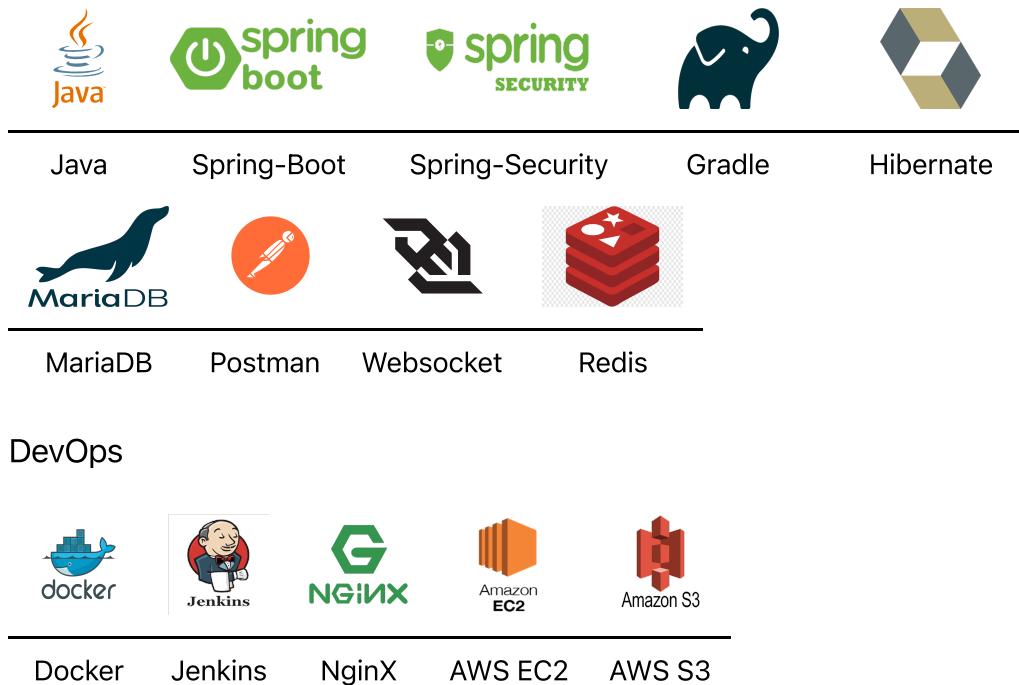
- 화상 기술을 통해 원거리에서 중고 상품을 확인할 수 있습니다.
- 판매자는 수요에 맞는 가격을 제시받고, 구매자는 가격에 맞는 상품을 구할 수 있습니다.
- 중고거래를 통해 일만큼 환경 보호에 기여하는지 알 수 있습니다.

## 2. 기술 스택

FrontEnd

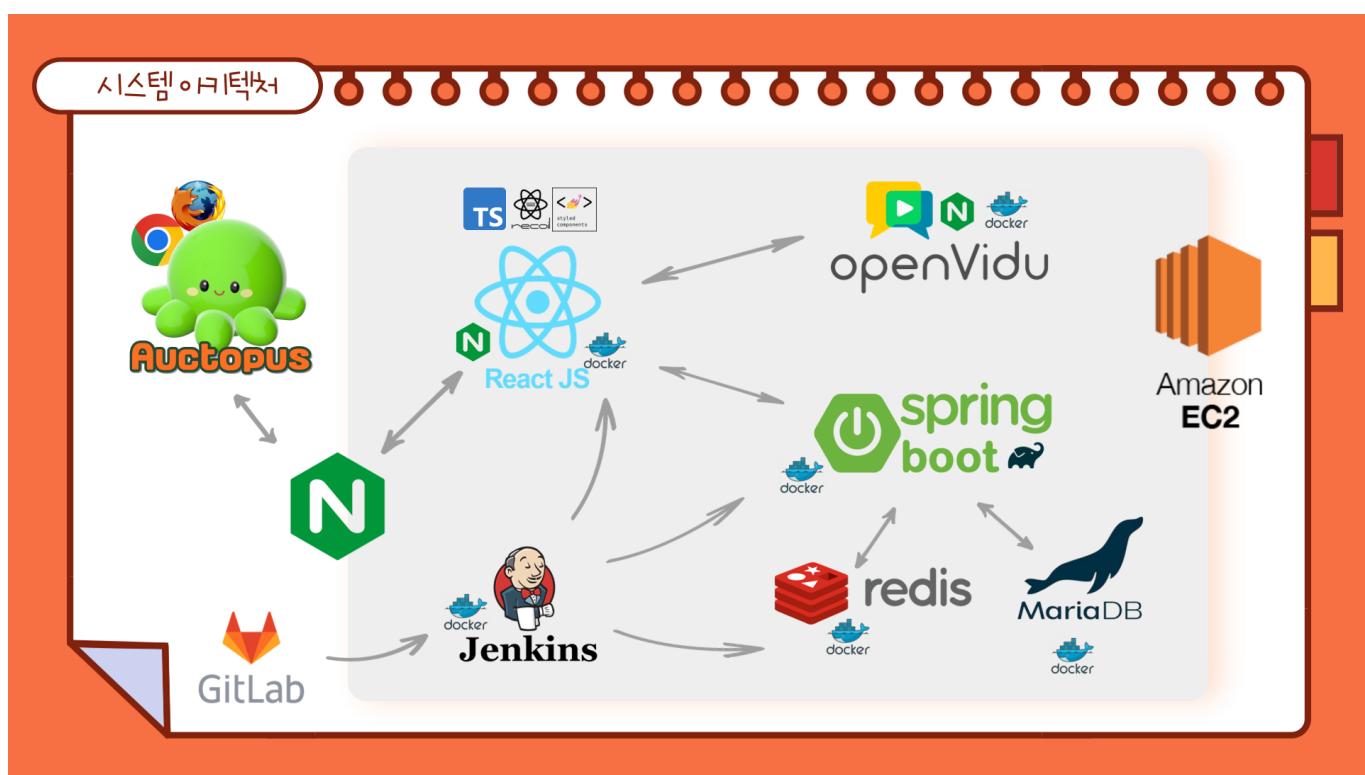


## BackEnd



## DevOps

### 3. 한눈에 보는 시스템 아키텍처



### 4. 팀원 소개

정상기 (팀장): Frontend 개발 + 프로젝트 관리 손예지 (팀원): Frontend 개발 + 디자인 김태원 (팀원): Frontend 총괄 + QA 이지원 (팀원): Backend 총괄 + QA 변유정 (팀원): Backend 개발 + 이슈 관리 우상빈 (팀원): Backend 개발 + 이슈 관리 + 발표

### 5. Jira Convention

Jira는 개발 이슈 트랙킹 용도로 사용합니다.

- 계층
  - 1. ⚡ (epic)
  - 2. 🖥 (FE), ✅ (BE), 🛠 (bug)
- 이름 규칙
  - Epic은 대분류입니다. FE, BE, bug 작업의 부모 항목입니다.
  - FE, BE 작업은 각각 프론트엔드 작업과 백엔드 작업 세부 이름입니다. (한글로 상세히 작성해 주세요)
  - 예시) 🖥 [FE] 유저 로그인 레이아웃 제작
- 이슈
  - 담당자, 우선순위, story point, Epic Link와 함께 이슈를 생성하세요
  - Backlog 탭에서 이슈를 미리 만들어 놓고, 매주 월요일 아침 회의를 통해 이슈를 개인별로 할당합니다. 할당한 이슈를 스프린트에 옮겨놓으세요
  - 해당 이슈를 완료하거나 작업을 시작했다면, Active sprints 탭에서 workflow (Todo, In-progress, Done) 을 변경해주세요
- Burndown Chart
  - 번다운 차트는 스프린트별 시간당 잔여 Story-point 를 추적합니다
  - 주차별 스프린트를 시작하기 전에 모든 이슈를 생성하고 할당해야 올바른 번다운 차트를 생성할 수 있습니다

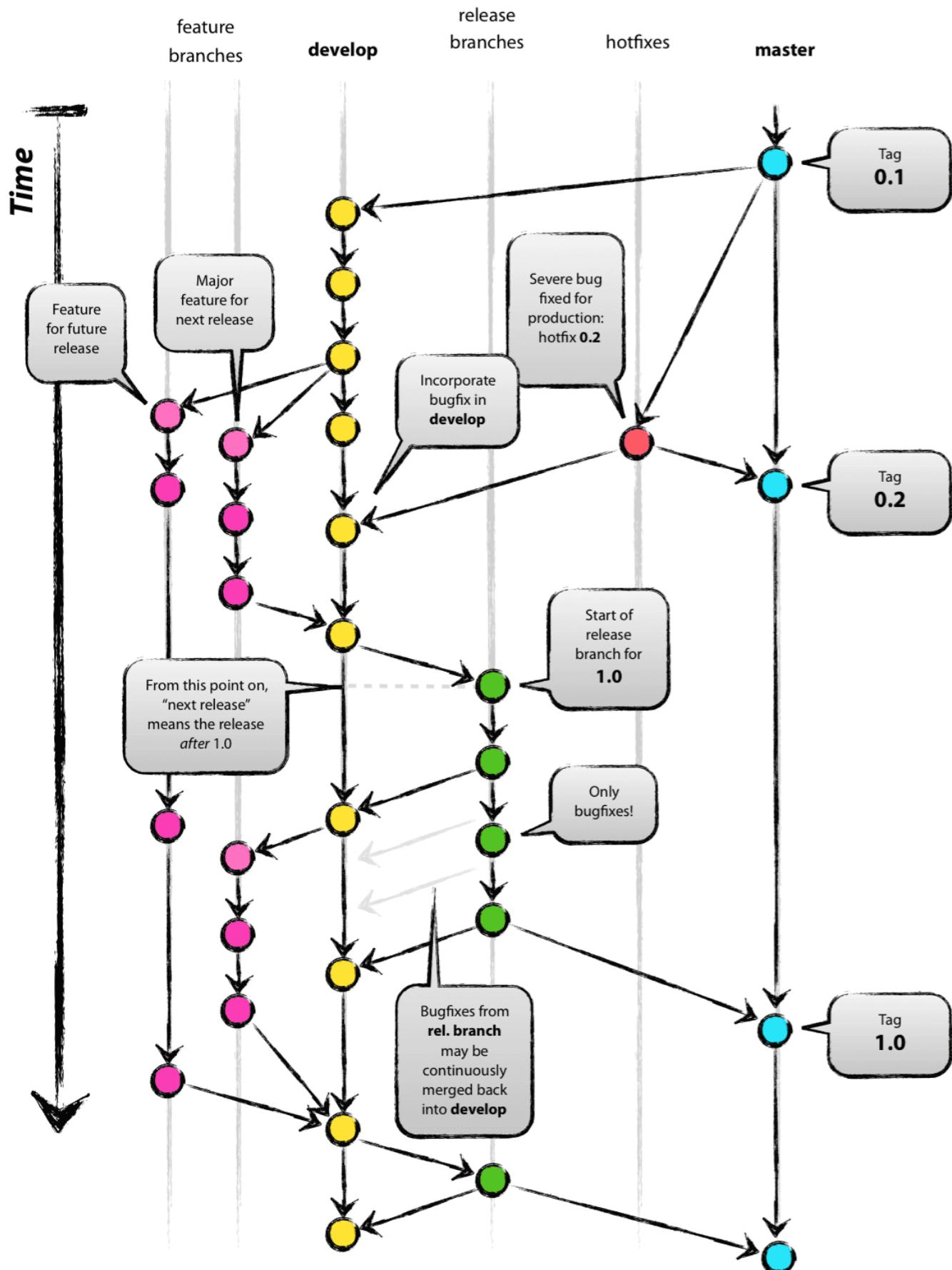
## 6. Git Flow Convention

디렉터리 구조는 **frontend** 와 **backend** 로 구분됩니다. 각 개발자들은 해당 디렉터리로 이동한다음 작업하면 됩니다

```
└── frontend  
    └── backend
```

### Branch Types

- main: 제품으로 출시될 수 있는 브랜치
- fe-develop: Frontend 개발 브랜치
- be-develop: Backend 개발 브랜치
- fe-feature: Frontend 기능 개발 브랜치
- be-feature: Backend 기능 개발 브랜치
- fe-hotfix: Frontend 버그 수정 브랜치
- be-hotfix: Backend 버그 수정 브랜치



1. origin/develop 브랜치에서 작업 브랜치를 생성합니다

```
(fe-develop)]$ git fetch
(fe-develop)]$ git checkout -b fe-feature/user-login-layout --track
```

```
origin/fe-develop
```

2. 작업 브랜치에서 작업합니다

3. 작업 브랜치에서 소스를 커밋합니다

```
(fe-feature/user-login-layout)]$ git commit -m "feat: add user login layout"
```

4. 작업 브랜치를 origin/develop에 rebase 합니다

```
(fe-feature/user-login-layout)]$ git pull --rebase origin fe-develop
```

5. 작업 브랜치를 origin에 push합니다

```
(fe-feature/user-login-layout)]$ git push origin fe-feature/user-login-layout
```

6. Gitlab에서 작업 브랜치를 develop에 합치도록 Merge Request 를 생성합니다

7. 동료에게 승인받고, merge 합니다

## 7. Commit Message Convention

### Commit Message Types

- fix: 내 코드베이스에서 발생한 버그를 수정했을 때
- feat: 새로운 기능을 추가했을 때
- docs: API를 바꿨을 때
- : docs: 문서 수정
- refactor: 코드 리팩토링했을 때
- test: 테스트 코드 추가 또는 수정했을 때
- style: 코드 형식 바꿨을 때
- chore: 그 외 기타사항
- fix 와 feat 외 타입들
- Examples:

```
~/SSAFY/my-app (be-feature/login) $ git commit -m " feat: allow provided config object to extend other configs"
```

```
~/SSAFY/my-app (be-feature/login) $ git commit -m "feat: check if ID is duplicated"
```

```
~/SSAFY/my-app (be-feature/login) $ git commit -m "feat!: send an email to  
the customer when a product is shipped"
```

```
~/SSAFY/my-app (be-feature/login) $ git commit -m "docs: correct spelling  
of CHANGELOG"
```

## 8. Backend Convention

- Style Guide
  - [Google Style Guide for Java](#)
- Project Structure

```
// backend 디렉터리 안에서  
.  
└── main                                // java src 디렉터리  
    └── java  
        └── com  
            └── underAuction  
                ├── api                      // REST API 디렉터리  
                │   ├── controller  
                │   ├── request  
                │   ├── response  
                │   └── service  
                ├── common                   // 공용 유ти 디렉터리  
                │   ├── auth  
                │   ├── exception  
                │   │   └── handler  
                │   ├── model  
                │   │   └── response  
                │   └── util  
                ├── config                  // 플러그인 설정 디렉터리  
                └── db                      // 디비 디렉터리  
                    ├── entity  
                    └── repository  
resources                         // 웹 리소스 디렉터리  
    └── README.md  
    └── application.properties
```

## 9. Frontend Convention

- Style Guide
  - [Google Style Guide for Typescript](#)
  - [gts](#)

- Project Structure

```
// frontend 디렉터리 안에서

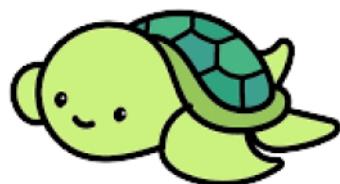
.
├── public          // 소스 코드에서 참조되지 않는 에셋
└── src             // 소스 코드
    ├── api           // 서버와 통신에 쓰이는 api
    ├── assets         // 소스 코드에서 참조되는 에셋
    ├── components     // 컴포넌트 디렉터리
    │   ├── Layout
    │   └── User
    ├── pages          // 페이지 디렉터리
    └── store           // Recoil 디렉터리
    types             // 타입 선언 디렉터리
```

## 10. Wire Frames

- 유저 화면 프레임들

9:41                          9:41                          9:41

### Auctopus 회원가입



아이디@이메일  
example123@gmail.com

비밀번호

비밀번호 확인

이름

닉네임

배송지 주소

거래 계좌 정보

**Login**

**Sign In**



9:41

9:41

## 비밀번호 재설정

## 관심 카테고리 설정

관심 카테고리를 설정해주시면  
해당 카테고리 우선으로 경매가 노출 됩니다.

이름

아이디@이메일

**임시 비밀번호 전송**

전자 기기

기타 음향기기

의류

고래 상어

정상기 화이팅!

변유정 화이팅!

김태원 화이팅

우상빈 화이팅

손지예 화이팅

A704 화이팅

이름과 이메일 계정이 일치하면 해당 이메일  
로 임시 비밀번호가 전송됩니다.

**건너뛰기****완료**

- 메인 화면 프레임들

9:41

진행중

진행 예정

진행예정인 live 경매

Airpod 좋아요 순 ^

라이브 시작일 3월 23일 17시

입찰시작가 450,000원

25명

Airpods Max 스페이스

라이브 시작일 3월 23일 17시

입찰시작가 450,000원

25명

Airpods Max 스페이스

라이브 시작일 3월 23일 17시

입찰시작가 450,000원

25명

초록 무너 좋아요 수호자 34

프라다 가방 팝니다. 한정판이에요

페션 침화

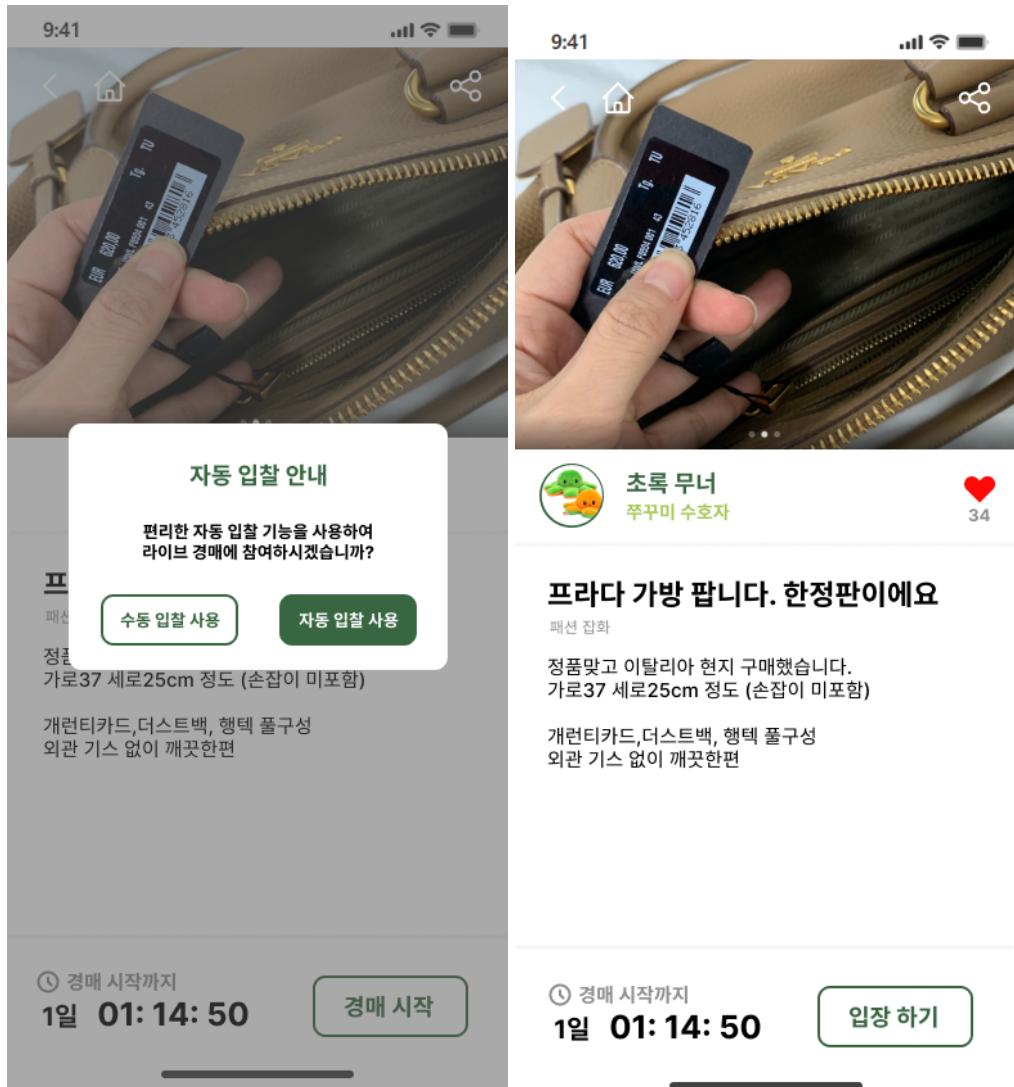
정품맞고 이탈리아 현지 구매했습니다.  
가로37 세로25cm 정도 (손잡이 미포함)

개런티카드, 더스트백, 행택 풀구성  
외관 기스 없이 깨끗한편

Airpods Max 스페이스 +

Home Chat Likes Profile

경매 시작 1일 01: 14: 50



- 하위 메인 화면 프레임들

9:41

9:41

9:41

9:41



정개미님의 관심목록



정개미님의 관심목록

진행 예정

입장 예정

**Airpods Max 스페이스**

라이브 시작일

3월 23일 17시

입찰시작가 450,000원

**Airpods Max 스페이스**

라이브 시작일

3월 23일 17시

입찰시작가 450,000원

**Airpods Max 스페이스**

라이브 시작일

3월 23일 17시

입찰시작가 450,000원

**Airpods Max 스페이스**

라이브 시작일

3월 23일 17시

입찰시작가 450,000원



Home



Chat



Likes



Profile



Home



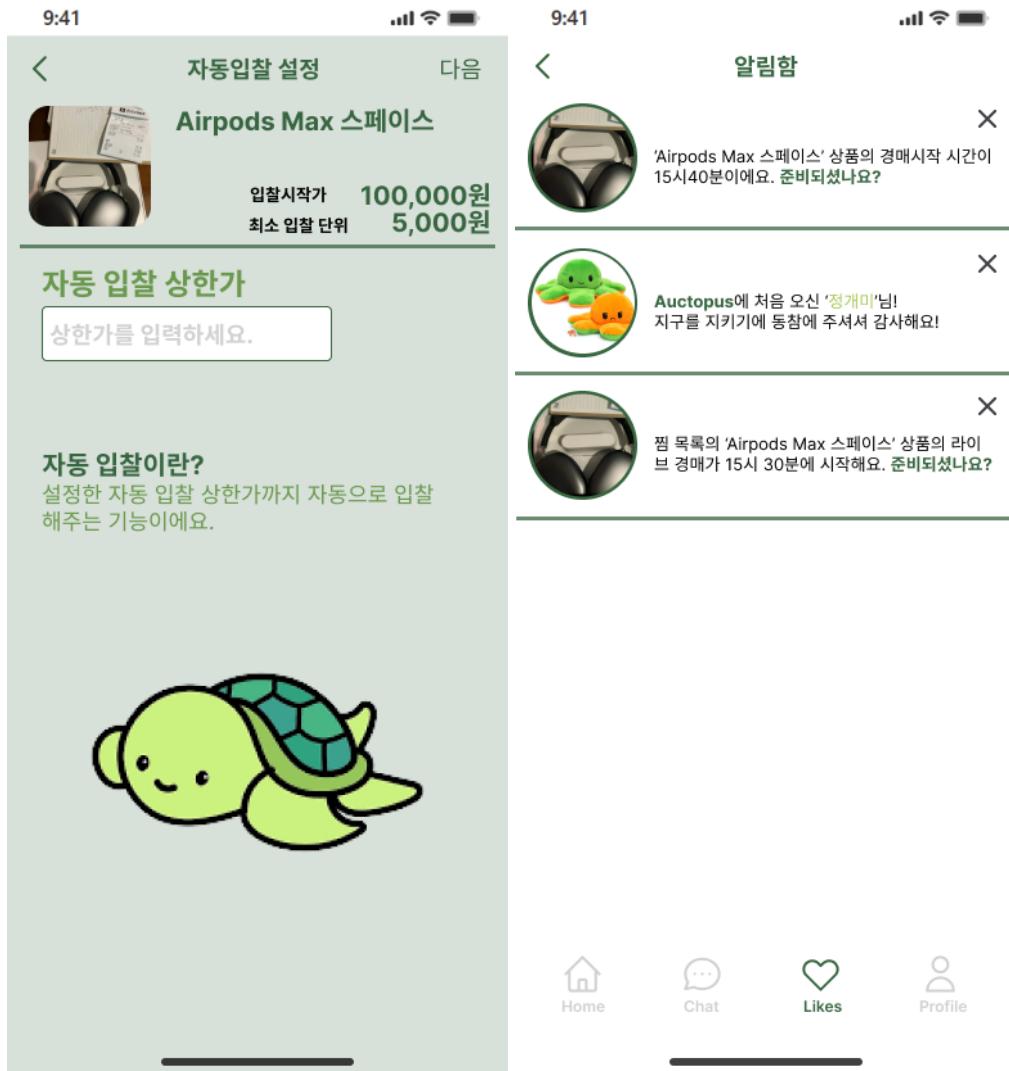
Chat



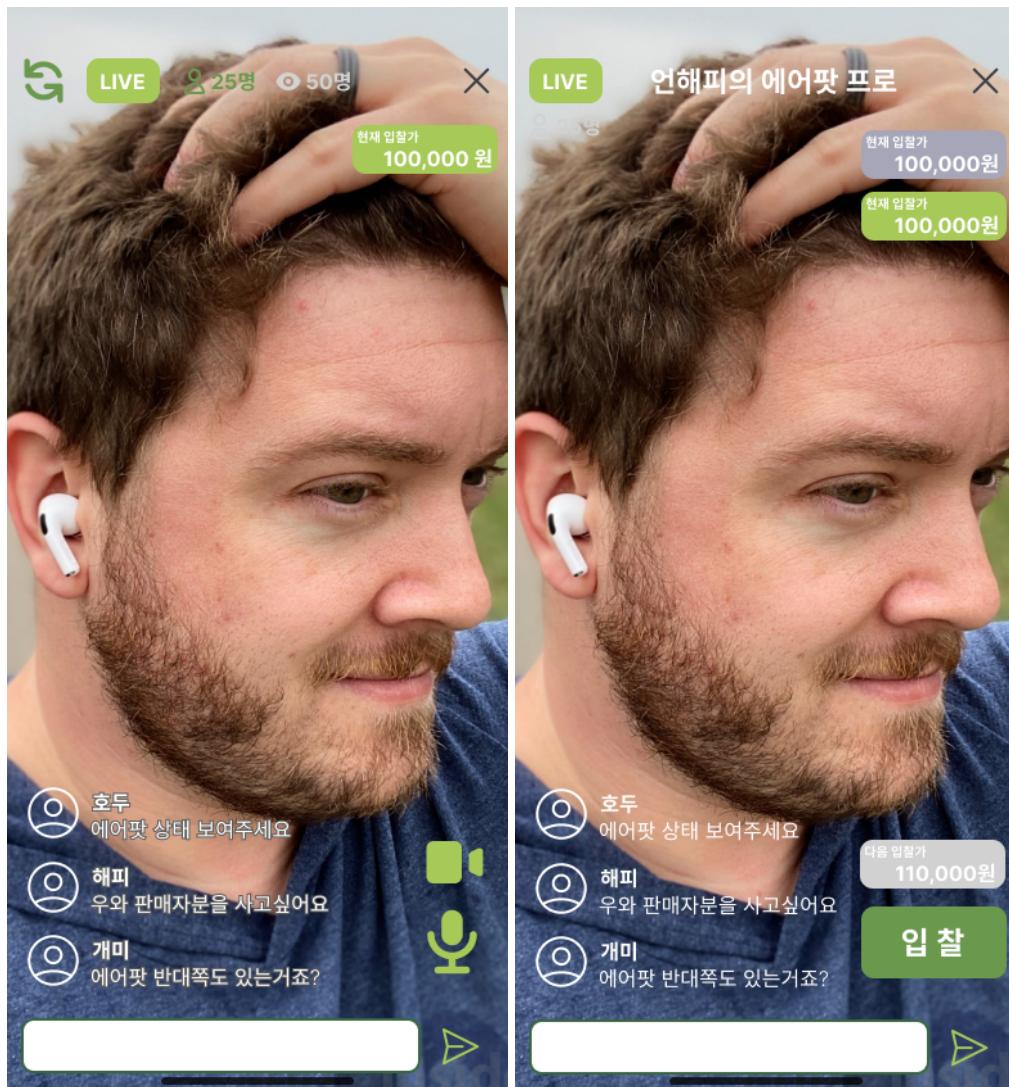
Likes

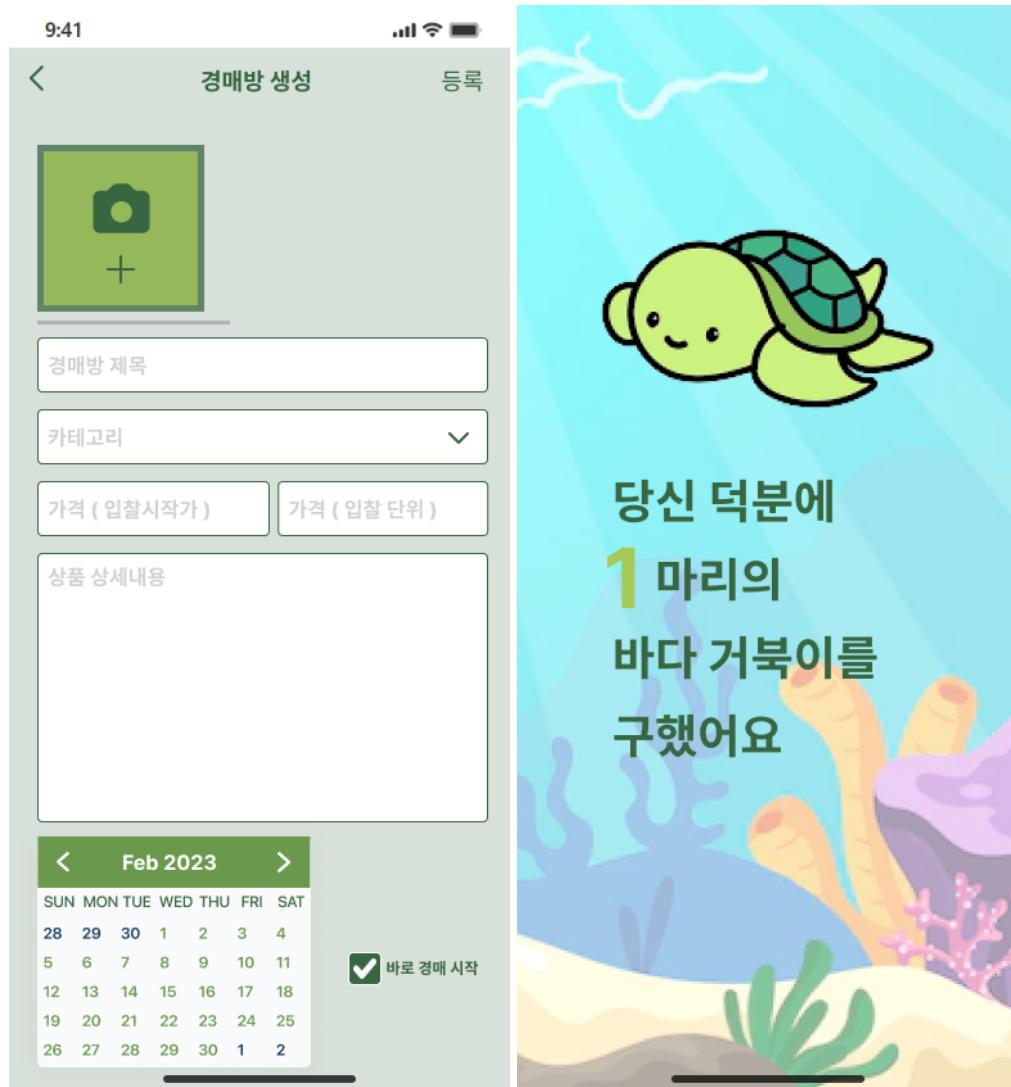


Profile



- 경매 화면 프레임들

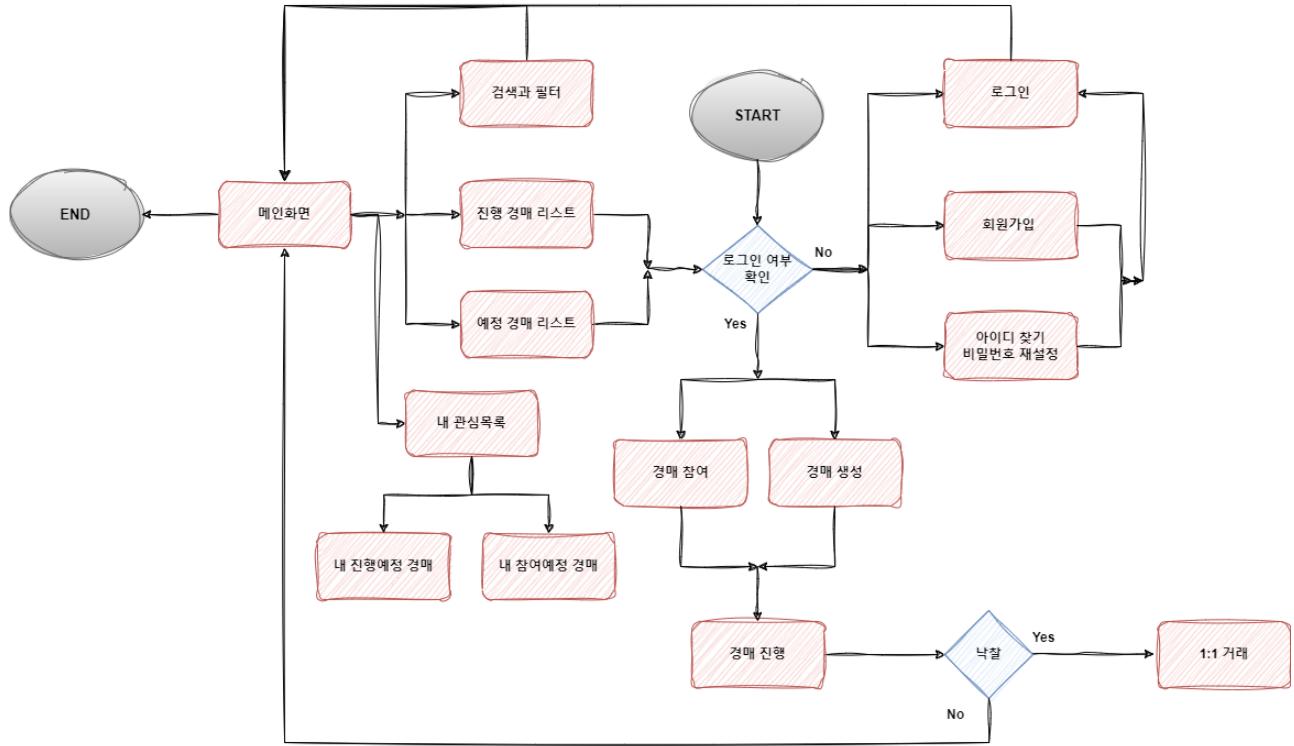




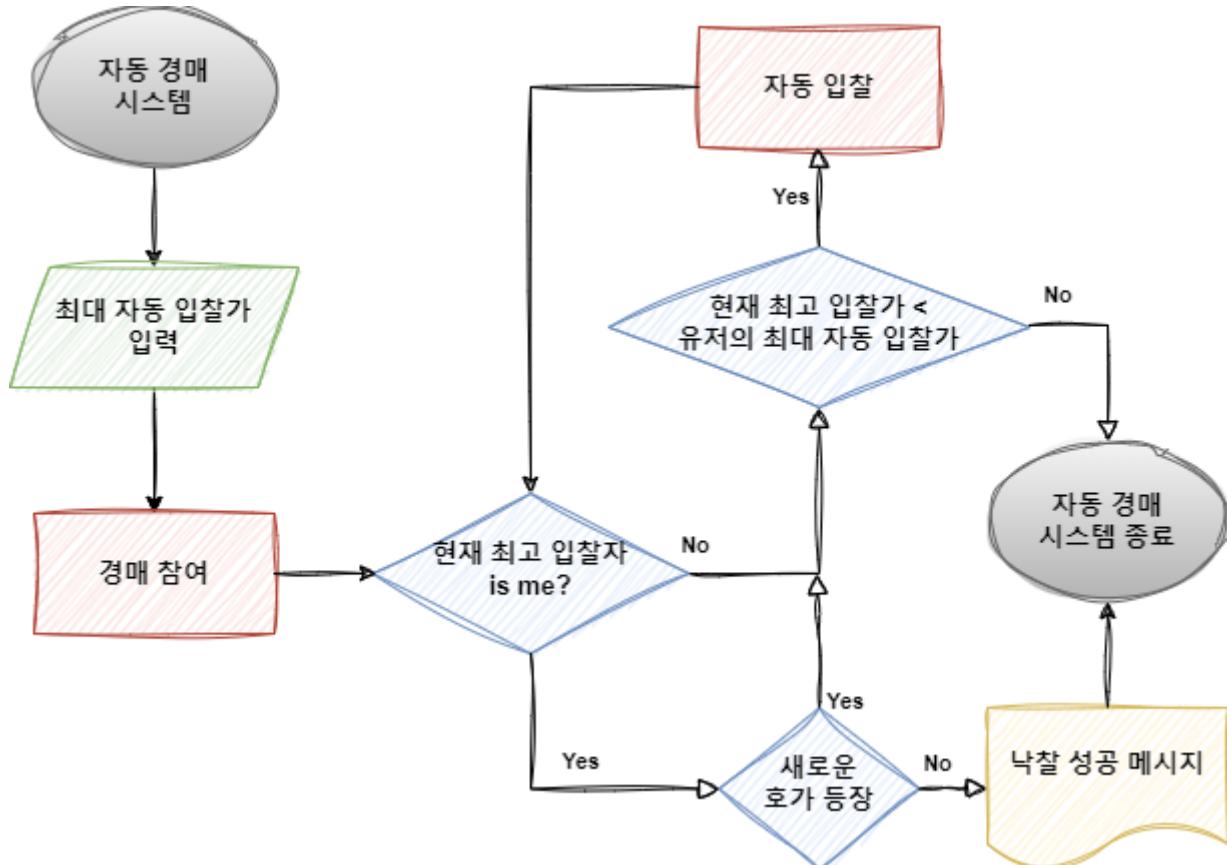
## 11. 기능 정의서

- 전체적 흐름도

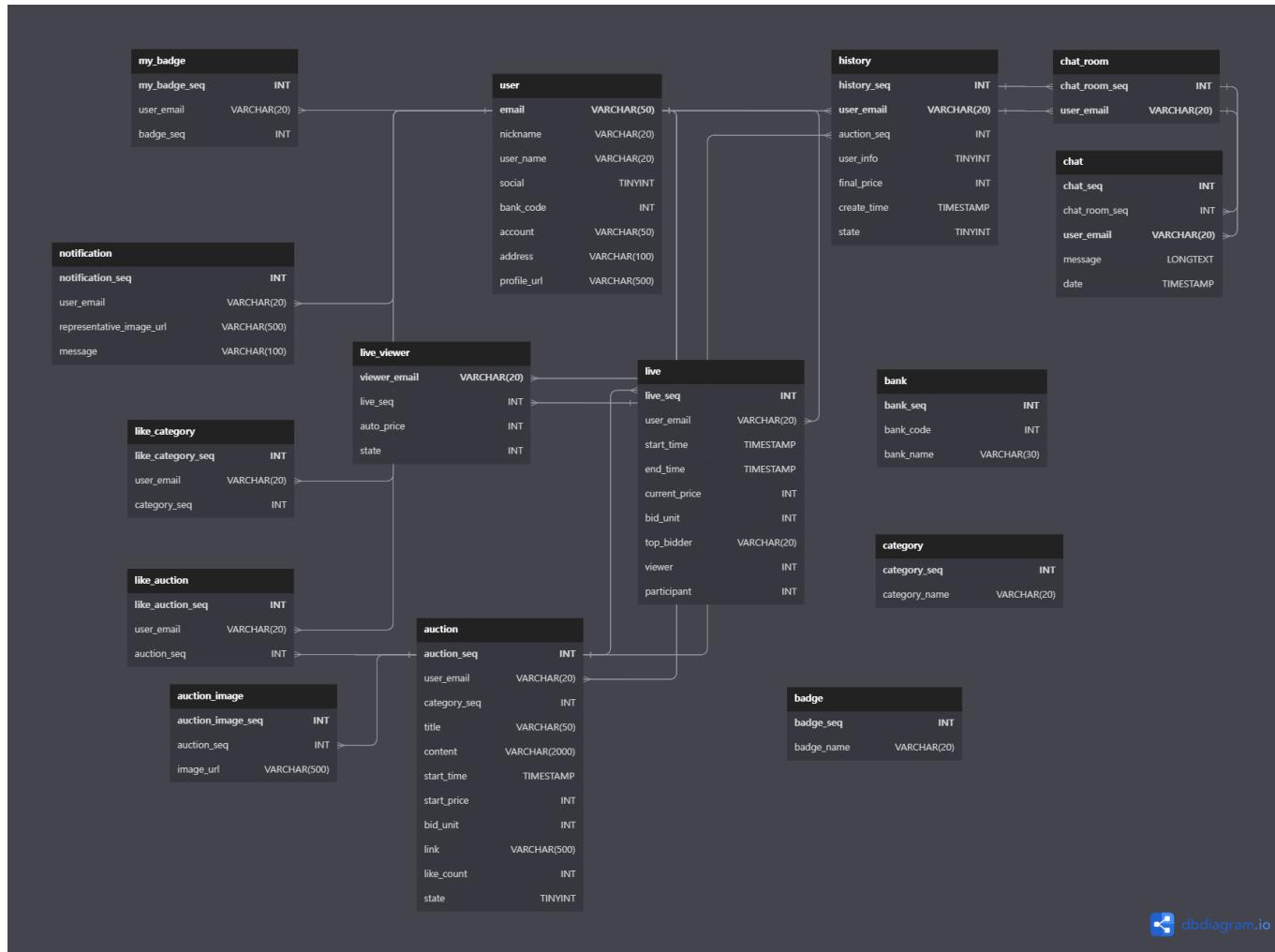
프로그램 전반적인 흐름도



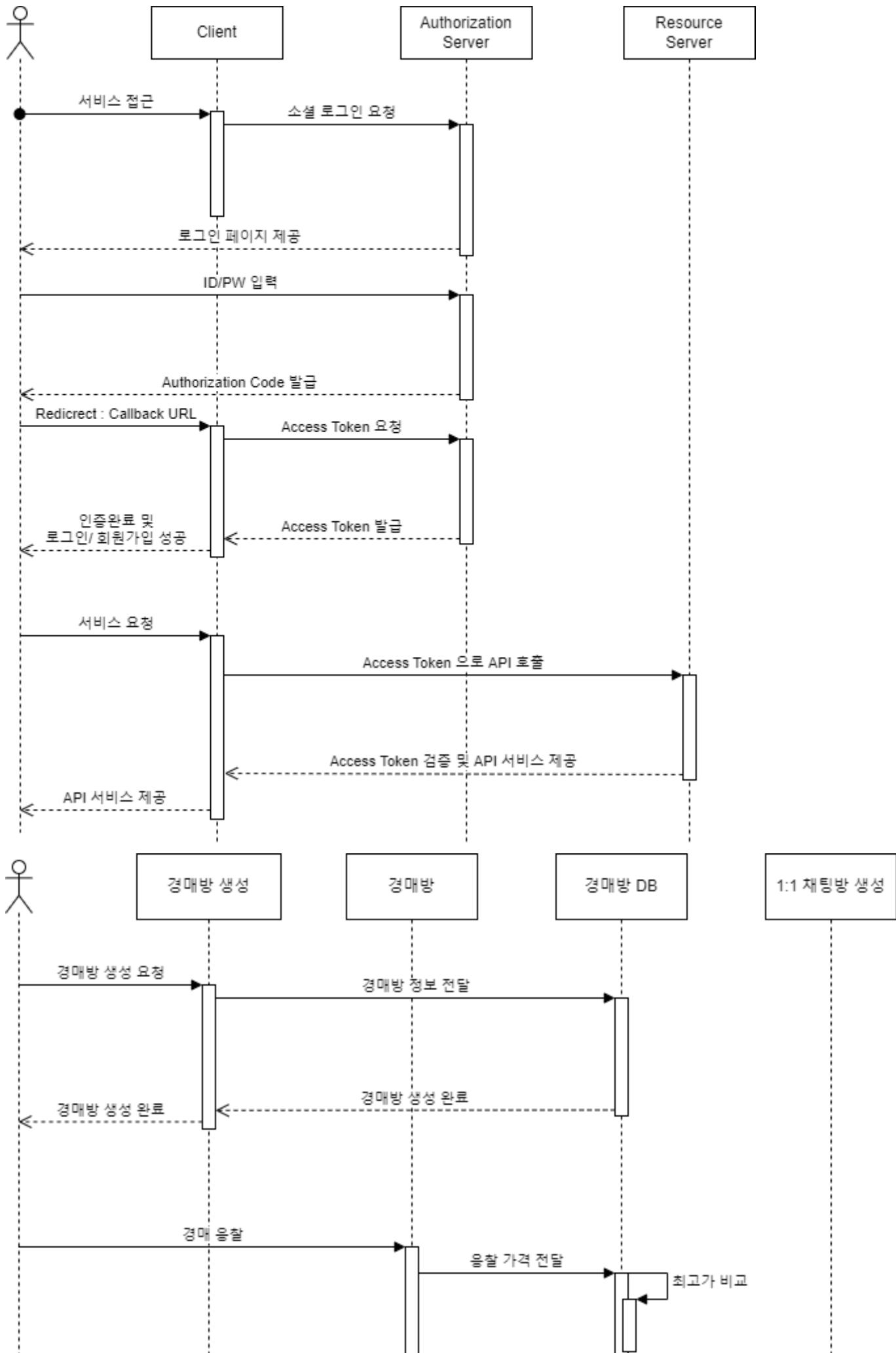
- 자동 경매 흐름도

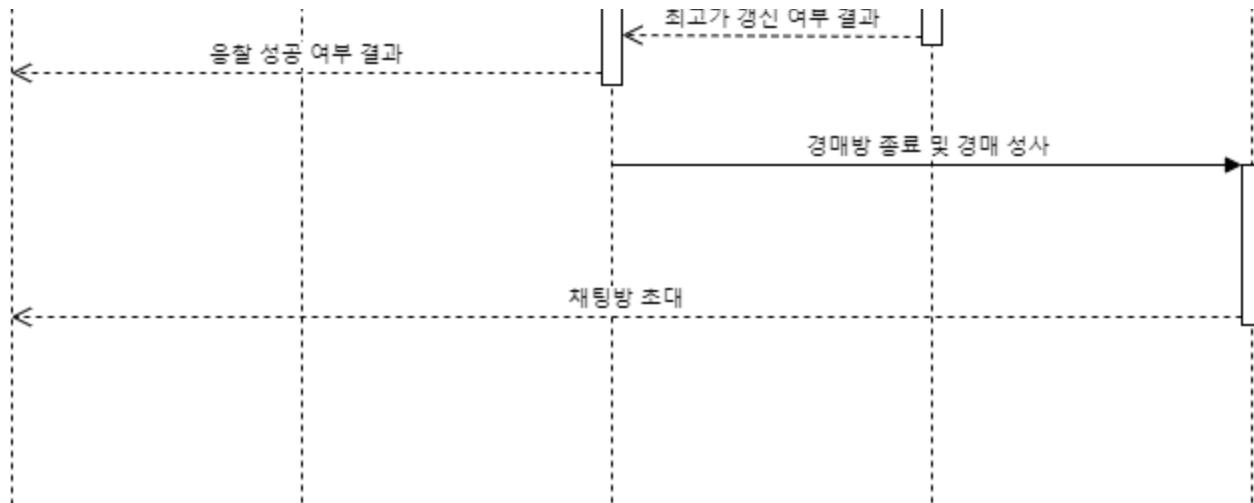


## 12. ERD



## 13. Sequence Diagram





## 14. References

- [우린 Git-flow를 사용하고 있어요](#)
- [Conventional Commits](#)
- [Google Style Guide for Java](#)
- [Google Style Guide for Typescript](#)