

# Essential Function

(E)

## I. 추천 기능



(4 stars)

☞ 확장성이 넓어야 요구됨

module화가 필수적임 !!!

① 옵 추천 : 선택을 기준으로 한 추천

② seller 추천 : 내가 좋아하는 seller의 비슷한 seller를 추천.

normally : → 구글 Basic하게 간단히 각 물품들의 category를 정하고 Random으로 뽑아주는 형식이 될 수 있음.

But, 이것은 불결국적으로 추천 서비스 아님, 추천 서비스가 project의 main function인 것을

Optional ① : 한가지 알고리를  
선택하면 추천 알고리를 많이 필요함.

Optional ② : 두가지 ✓ 옵 추천 Content Based Filtering : 사용자의 Item에 대한 Data를 가지고  
선택한 유저의 이어짐이나 / 나와 비슷한 유저의 사용자

☞ Upn의 module 존재함.  
✓ Seller 추천 Collaborative Filtering : 내가 몇번 평균 rating을 가지고  
좋아하는 이어짐

나와 비슷한 사람의 선호하는 이어짐을 추천

(S)

## II. 결제 기능



→ 가능 과정은 어렵지 않음, Open API와 결합성이 있음.

(上)

### III. 경매 기능

Optional ①

→ 미리작동하기 때문에 Ram 필요

① 캐팅기능 : client server와 seller server의 Socket 통신 필요함.

시간 관리 도구들

借此 배우면 '도입식별' ② 시간 관리 : nomaly : 단순히 client 단에서 해당 시간이 되면 request를 보내기 풍으로

같은 기능에서 코드 재사용률을 높일 수 있음. Optional 1 :

Server 에서 해당 시간이 되면 Api를 차단 (이용 보안)

- SSE로 서버 시간 받아오기, Socket으로 오류에게 공유하기

- node-schedule를 이용해 낙찰자.

(上)

### IV. Database modeling

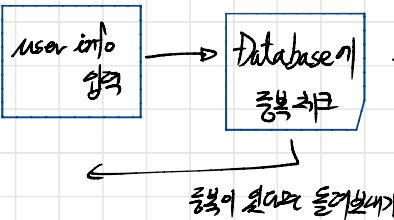
※※※※※

→ Server 개발자들의 이야기. 매우 흥미롭음.

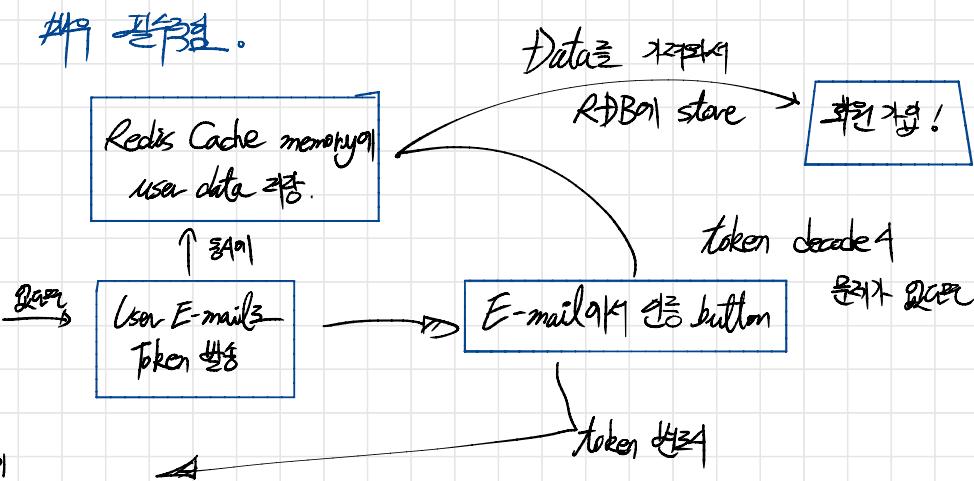
(中)

### V. 회원가입 & Login 가능

① 회원가입



Redis Cache memory  
user data 등록.



Opinion 1. 카카오 택시의 같은 application처럼

client

판매 / 구매에 관한 것은 하나의 application으로 만들고

관리자는 admin page를 만드는 것을 고려해 볼 수 있다.

서비스하는 판매자, 구매자, 통합 회원가입 API는

MSA로 분리하여 관리할 필요가 있을

Opinion 2. 상용 chatbot 도입의 여지가 있을

기존의 Socket API 채팅 기능이 구현되기 때문에 상용 채팅 도입은 어렵지 않을

최적화로 빠른 가능에서 img를 보여줄 수 있는 기능이 필요할 + MySQL 사용.

# Version 관리

# Refactoring 힌트 .

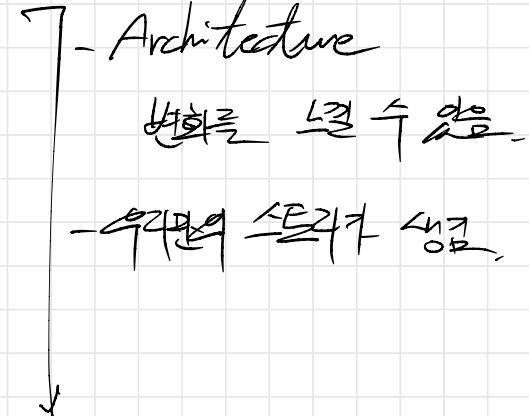
- SemVer ~~이거로~~ version 관리.

① 기존의 방법대로 (code 풀기, module화 고려하기)

minor  
version  
up - monolithic architectures 주제

② Design Pattern 적용

③ UML을 아키텍처 틈



\* Version Up이 같은 chapter Realize Note 주제.

# Essential Requirement

- ① Javascript의 공통적인 Convention을 지킬 것.
- ② ES6 문법을 사용할 것.
- ③ code review는?

Value Proposition.