



# 퓨처 포테이토 PMP

---

팀장 박신형

팀원 김다빈

팀원 박형우

팀원 이유성

# Future Potato

## PMP INDEX

캠프를 통해 이루고 싶은 것은 무엇인가요?

01.

### 목표

—

- 1.1. 개인 목표
- 1.2. 팀 목표

진행하는 프로젝트의 방향은 무엇인가요?

02.

### 프로젝트 구성

—

- 2.1. 프로젝트 선정계기
- 2.2. 기능 구성
- 2.3. 기능별 개발 계획

어떻게 프로젝트가 흔들리지 않게 관리할 것인가요?

03.

### 협업 방식

—

- 3.1. 위험측정
- 3.2. 극복 방식

## 1.1. 개인목표

김다빈 – 박신행 – 박형우 – 이유성 순으로 진행됩니다.

## 1.1. 개인목표 - 김다빈 Mobile

구상계기 : 현재 상황의 문제점

01 코드 수정의 어려움

02 복잡하고 비효율적인 코드

-> 코드 이해의 어려움

목표 : 해결방안

아키텍처 패턴 적용하여 유지관리를 개선하고  
일관성을 부여하여 코드의 가독성을 높이자!

: 어떤 아키텍처?

### MVVM 적용하기

MVC 패턴은 구현이 쉽지만, 기능 추가, 변경에 있어 유지보수가 어렵다는 단점,  
앱 동작이 많아질수록 Activity가 무거워진다는 단점이 있다고 하여, 이를 보완한 MVVM패턴 선택.

## 1.1. 개인목표 - 김다빈 Mobile

### 아키텍처를 적용하기 위한 방법

#### 01 MVVM 구조 파악하기

현재 이론은 알지만 어떤 식으로 구현해야할지 모르겠는 상태  
예제 코드 참고하여 MVVM 구조를 분석하고 흐름 파악할 것

#### 02 구현할 기능 구조화 하기

이해하고, 분석한 내용 바탕으로 구조짜기  
-> 뷰- 뷰 모델-모델 각 영역 구분하여 구조화

#### 03 구현하기

구조화한 것을 바탕으로 구현해서  
예상대로 동작하도록 하는 것이 목표

### 추가적으로

코드를 완벽히 이해하기 위한 노력의 필요성

#### 01 매일 개발 일지 작성

구현한 코드 복습하기 위해 기록  
(내 코드를 완벽히 이해하고 있는 상태를 유지하기 위함)

#### 02 기록 습관화

구체적이고 간결하게 문서화 작업  
⇒ 포트폴리오까지 이어지도록!

## 1.1. 개인목표 - 박신행 BE

### 구상계기 : 도전 욕심

01

검색/추천 BE 쪽 JD 주요업무에 나와있는 데이터수집에 도전하고 싶음.  
-> SNS에서의 데이터활용의 대표적 예시는 추천시스템이므로,  
그중 추천팔로잉 기능을 만들기 위한 데이터 수집에 도전하고싶음.

02

게시글 취향을 고려하지 않고 팔로잉 정보만 고려하는  
기존 인스타그램의 추천팔로잉 기능 개선하고 싶은 욕심

### 목표 : 해결방안

추천 팔로잉의 **기반이 될 수 있는** 로그 남기기  
(추천 팔로잉 개발까지 가지 않음 - 시간부족)

### “기반”이 되기 위해 어떤 것을?

**01** 어떤 유저의 행동이 분석에 의미 있는지 분석

- ex. 페이지 방문 시각, 좋아요 유무

**02** 원하는 결과가 나올 수 있을 것 같은  
이벤트 로그 설계 사례 탐색 (Tracking Plan)

-이벤트 이름, 어느 시점에 실행  
(ex.이름- 파라미터까지 포함 저장하는 경우 vs. 이벤트만 추상화)

### 의미가 있었는지 어떻게 체크?

- 유저 유형을 나눌 수 있는지 확인  
- 사이가 가까운 인물을 구분할 수 있는지 확인

## 1.1. 개인목표 - 박형우 BE

### 구상계기 : 현재 상황의 문제점

#### 01

DB설계 없이 개발을 바로 시작하여  
개발 도중에 전체적인 구조를 파악하기 힘들.  
이로 인해 개발흐름을 제대로 이해하기 어려움.

#### 02

수정이 필요한 경우  
어떤 테이블을 건드려야할 지 감이 잡히지 않아  
막막함을 경험하였음.

### 목표 : 해결방안

#### 테이블간 **명확한 DB설계**

### “명확하게” 하기 위해 어떤 것을?

#### 01 테이블 목록 작성

요구사항(필드의 데이터 타입, 유효성 검사)을  
만족하는 테이블 목록 작성

#### 02 ERD(관계도) 그리기

테이블 간의 카디널리티 비율을 파악하여  
ERD(관계도) 그리기

### 기대효과 : 추가적으로 내가 얻을 수 있는 것은...

엔티티간의 연관관계를 한 눈에 볼 수 있음.  
-> 이를 통해서 데이터 흐름이 어떻게 이루어 지는지 파악할 수 있음

## 1.1. 개인목표 – 이유성 FE

### 구상계기 :

#### 컴포넌트의 재사용성 이해하기

- 평소에 작은 컴포넌트부터 확장하여 개발하는 방식, 즉 컴포넌트를 재사용하는 패턴의 장점을 체감하고 싶었음.
- Atomic pattern을 사용하면 애플리케이션과 분리해서 컴포넌트를 개발할 수 있고 이는 컴포넌트의 재사용성을 보다 자세하게 이해할 수 있을 것 같아 목표로 설정

### 목표(1) : 해결방안

Atomic pattern을 사용하여 코드 작성하여 컴포넌트의 재사용성을 극대화하기

### 재사용성을 이해하기 위해 어떤 것을?

#### 01 화면 분석하기

화면을 분석하여 HTML TAG단위로 구분.

Ant Design을 사용할 계획이 있으니 ant design을 사용하는 부분을 제외하고 구분 및 분석.



## 1.1. 개인목표 – 이유성 FE

재사용성을 이해하기 위해  
어떤 것을?

### 02 Atomic 패턴 사용

Atoms, Molecules, Organism, Templates, Page를 차례로 구성함.  
이때 각각의 구분이 모호해질 수 있으니, 주에 한 번씩 확인하면서 각각 구분을 할 수 있게 코드를 작성함.

#### 01. Atom

HTML TAG단위로 작성  
재사용이 가장 자주 일어나는 만큼  
Styled Component 사용

#### 02. Molecules

Atom 단계에서 구현된  
컴포넌트로 구성  
만약 구성 중 필요한 Atom이  
있다면 구현 후 조합

#### 03. Organism

컴포넌트가 윤곽이 보이게  
Molecules와 Atom을 조합

#### 04. Template

구현한 Organism과  
ant design을 배치하여  
layout을 구성

기대효과 : 추가적으로  
내가 얻을 수 있는 것은...

Atomic pattern을 익힘으로써 다른 pattern을 익힐 때 더 빨리 익힐 수 있음.

## 1.1. 개인목표 – 이유성 FE

### 구상계기 : 현재 상황의 문제점

기존의 CSS파일을 따로 두는 방식에 단점

- CSS 파일을 따로 두고 설계하니  
프로젝트 도중 CSS 수정 시 컴포넌트에 해당하는 CSS 찾기가 어려움
- Classname을 중복해서 하는 빈도가 많았고,  
원하지 않는 컴포넌트에 CSS가 적용되는 빈도도 많았음

### 목표(2) : 해결방안

Styled component를 사용하여 CSS 스타일링 하기

### Styled component를 활용

#### Styled component 활용

CSS 파일을 JS안에 사용하는 styled component를 사용함.  
Styled component를 사용하되 최대한 기존의 컴포넌트 styled을 확장,  
혹은 props를 사용한 스타일링을 할 수 있도록 함.

### 기대효과 : 추가적으로 내가 얻을 수 있는 것은...

1. Props와 기존의 컴포넌트를 활용하여 스타일을 할 수 있기 때문에 간단하게 동적 스타일링을 할 수 있음.
2. 다른 사람에게 설명할 수 있을 정도로, 또 이를 다른 프로젝트에서 활용할 수 있을 정도로  
Styled Component 스타일링 방법을 깊이 있게 안다고 할 수 있음.

## 1.2. 팀 목표

## 1.2. 팀 목표



### 팀 위키 제작 with Notion

각자 보완하고 싶은  
개발스타일을 문서화를  
통해 극복하자!

## 구상계기

### 01 개념에 근거한 코드에 대한 필요성

시간이 촉박하니 결과물이 급해서, 동작원리에 대한 이해는 제쳐두고(각기 다른 아키텍처를 고려하지 않고) 무작정 코드를 복붙하여 만드는 경우가 종종 있음  
→ 코드 충돌 문제 발생  
→ 왜 에러가 터졌는지 해결하는데 오랜 시간이 걸림

### 02 같은 실수 반복 방지

문제 해결에만 급급하여 오류발생 시 정확한 원인을 파악하지 않고 구글링을 통한 복붙으로 해결함.  
→ 똑같은 상황을 마주하였을 경우, 무엇이 문제인지 알지 못하여 에러를 해결하지 못하고 또 다시 구글링을 하여 시간적 낭비가 발생함.

## 1.2. 팀 목표

### 목표 및 구체적 행동

#### 01 개념에 근거한 코드에 대한 필요성

##### - 김다빈

어떻게 짜야할지 머릿속에 그려져서,  
(구글링 없이) 스스로 코드를 짤 수 있는 상태가 되기

##### 1. 샘플 코드를 직접 따라해보면서 실습

- 여러 번의 실습 과정으로 과정을 정확하게 이해해야 할 것.

##### 2. 이후 그 (학습) 과정을 리마인드 하기 위해 정리

- 따라하지 않고, 혼자 구현할 수 있는지 체크함으로써 목표 달성을 측정함

##### - 박신행

(우리가 세운) 아키텍처를 이해한 사람에게  
데이터의 흐름을 설명할 수 있을 정도로,  
동작원리 개념에 근거하여 코드 작성하기

##### 1. 개인블로그글보다 다수를 위하고 신뢰성 있는 자료 참고(=책)

- 동작원리에 대해 정확히 이해함

##### 2. 추가로 정기회의 코드리뷰 시 모든 팀원이 데이터의 흐름을 이해할 수 있게 설명하기

- 모두 이해했는지 체크함으로써 목표 달성을 측정함

##### - 이유성

타 파트의 코드를 이해하는 데  
오랜 시간이 걸리지 않은 상태가 되기

- 개발 경험이 부족하므로 다른 파트의 지식 또한 쌓고 싶고자 함

##### 1. 다른 팀원들의 위키 보며 관심가는 키워드를 바탕으로 확장한 개념 정리

##### 2. 다른 팀원의 위키에서 설명이 부족한 부분 보충

## 1.2. 팀 목표

### 목표 및 구체적 행동

#### 02 같은 실수 반복 방지

##### - 박형우

오류 발생 원인 분석을 통한 문제 해결능력 향상

- 구글링을 통해서 어떤 상황에 어떠한 오류가 발생하는지를 파악  
→ 왜 오류가 났는지 + 어떻게 해야 해결할 수 있는지 기록

## 1.2. 팀 목표



### 팀 위키 제작 with Notion

각자 보완하고 싶은  
개발스타일을 문서화를  
통해 극복하자!

### 기대효과

#### 01 속도는 느릴지 몰라도 해당 코드를 넣어도 될지에 대한 확신이 있음

- 개발 속도는 느릴지 몰라도 해당 코드를 넣어도 될지에 대한 확신이 있음
- 따라서 에러가 터질 확률이 줄어들고(코드 충돌 문제 방지)  
어디서 문제가 생겼는지 빠르게 파악 및 대처 가능.
- 협업뿐만 아니라 향후 기술면접에도 도움이 될 거라 생각

#### 02 같은 실수 해결방법 참고 가능

이후, 전에 발생했던 오류를 마주할 경우

- 왜 오류가 났는 지 이유를 알기에 빠른 디버깅이 가능
- 해결방법이 떠오르지 않아도 기록한 부분을 보며 즉시 해결을 할 수 있음
- 이후 같은 문제를 겪고 있는 팀원에게도 도움됨.

#### 03 회고가능한 데이터이자 포트폴리오

- 문서화 자체가 포트폴리오
- 각자 회고 기준에 맞춘 문서

## 2. 프로젝트 구성



## 2.1. 프로젝트 구상 계기

### 보안 태그 개발 : 개인정보 보호를 목적으로 한 해시 태그 개발

#### 구상계기 :

- SNS에 개인정보를 입력 시, 공유하고자 하는 사람들에게 모두 공유되었음에도 불구하고, 깜박 잊고 못 지우는 경우가 있어 개인정보 유출이 이루어짐
- 특히, 개인정보 교류가 많이 이루어지는 DM기능(메시지)에 개인정보 보호 측면에서 도움이 될 거라 생각
  - ex. 매장 홍보, 중고 거래

#### 팀원이 가장 많이 사용한 SNS는?

- Instargram
- 

#### 프로젝트 방향

SNS의 핵심 기능만 구현하고, **그 위에서 보안태그가 쓰일 수 있게 하자!**

## 2.2. 기능 구성

### 보안 태그 요구 사항 - FE

#### 태그 관리 페이지

##### 01 태그 목록 페이지 - 등록, 수정, 삭제

배송지 선택

박00

기본배송지

강원도 춘천시 우석로 000호

010-1234-5678

문 앞

수정

선택

박00

인천광역시 중구 운서동 0000-0 네이버 아파트, 000동

000호

010-1234-5678

문 앞

수정

선택

+ 배송지 추가

택배 주문 절차:  
배송지 정보 등록-> 저장했던 배송지 선택-> 배송지 지정

보안태그 관리 하기

회원님의 보안태그 목록을 확인하고 여러 경로를 통해 게시된 보안태그의 상태를 손쉽게 변경할 수 있습니다.

보안태그 기록

보안태그 목록

새로운 보안태그를 등록하여 저장 후 이를 다시 선택하는 식으로 사용할 수 있습니다.

또한 현재까지 등록된 보안태그의 목록을 확인하고 그룹으로 정리할 수 있습니다.

수정

삭제

그룹명	보안태그명	관리		
집주소	강원도 춘천시 석사동 000-0, 000호	수정	삭제	
<div>집주소 ▼</div> <div>전화번호</div> <div>집주소</div>	인천광역시 중구 운서동 0000-0 네이버 아...	등록	수정	삭제

+ 보안태그 추가

## 2.2. 기능 구성

### 보안 태그 요구 사항 - FE

#### 피드/댓글에서 태그 입력 UI

##### 01 피드 – 사용자 태그처럼 UI 구성



##### 02 댓글 – 아이콘으로 태그 입력할 수 있는 UI 구성

(이모티콘 대신 입력UI로)



## 2.2. 기능 구성




### 보안 태그 요구 사항 - FE

#### 태그 관리 페이지

#### 02 태그 기록 조회 페이지

저장된 비밀번호			추가	
사이트	사용자 이름	비밀번호		
 1365.go.kr	limo730	.....		⋮
 apply.42seoul.kr	limo730@naver.com	.....		⋮
 49.50.167.11:3000	201812708@kangwon.ac.kr	.....		⋮
 49.50.167.11:3000	limo730@naver.com	.....		⋮

크롬브라우저의 '비밀번호 관리' 기능  
: 자동완성으로 저장해둔 비밀번호를  
하나의 웹페이지에 모아 관리할 수 있는 기능

보안태그 기록		보안태그 목록
보안태그 기록		
보안태그가 어디에 사용되고 있는지 파악하고 개별로 삭제할 수 있습니다.		
그룹명	보안태그명	관리
집주소	강원도 춘천시 석사동 000-0, 000호	
집주소	인천광역시 중구 운서동 0000-0 네이버 아	
기본 그룹	010-1234-5678	
피드	졸업한지 벌써 2년이라는 세월이 흘렀네~ 대학교 생활...	<div>링크</div> <div>삭제</div>

## 2.2. 기능 구성

### 보안 태그 요구 사항 - BE

#### 01 태그를 이용한 정보 입력 (태그 등록)

- 태그에 입력될 개인정보 분류  
: 주소(우편번호, 상세주소) / 생년월일 / 이메일 / 이름 / 연락처 / 계좌정보(계좌번호, 예금주)
- 분류 설정 기준 : 개발자 대상 개인정보 보호조치 적용 가이드 문서(2020.12, KISA)

#### 02 설정 시간 지나면 자동 삭제 (타이머)

설정 시간 : 10분 / 1시간 / 6시간 / 1일 / 1달 / 1년 / 지정하지않음 / 사용자 지정

#### 03-1 피드/댓글에서의 태그 삭제 : (1) 태그 단위 삭제 (2) 전체 삭제

- 태그 삭제 시 태그 이외의 내용은 존재함.
- 태그 구분 : 피드id/댓글id + 개인정보id  
ex. 연락처정보(id=1)가 피드1(id=1)과 피드2(id=2)에서 사용되고, 각각 타이머 또한 다를 때  
태그1과 태그2는 동일하지 않은 인스턴스로 인식
- 태그 단위 삭제 : 태그별로 타이머를 다르게 설정 가능

#### 03-2 태그 목록에서의 태그 삭제 : (1) 태그 단위 삭제 (2) 전체 삭제

- 태그 목록 페이지에서 등록한 태그자체를 삭제하면 모든 게시글에서 다 삭제됨

## 2.3. Instagram 🍪

### 기능별 개발 계획



### 3. 협업 방식

## 위험요소 및 극복제도 (그라운드 룰)

### 3.1. 위험요소

#### 01 지쳐서/방심해서 미르는 경우

일정 소화에 차질

#### 02 일정만 세워두고 각자 개발하는 상황

각자 현재 무엇을 하고 있는지 공유 X  
-> 계속 DM으로 물어봐야 함  
/ 일정을 소화하지 못할 경우 원인 분석 어려움

#### 03 무기한적으로 에러를 붙잡는 경우

#### 04 하고 싶은 말 답아두는 경우

쌓이는 악감정으로 팀워크가 깨짐

### 3.2. 극복제도 -> 그라운드 룰

#### 코어 타임때 화면 공유 (11AM - 4PM)

디스코드로 화면 공유해서 함께 프로젝트 하는 분위기 조성

#### 1주 단위 진척 관리

1. 노선으로 일일 진척 공유(개발한 것 & 이슈체크 & 다음날 Todo)
2. 정기회의 (목) : 개발한 것들 공유 & 이슈체크 & 차주 Todo 공유
3. 개발일지 (토) : 1주동안 팀의 진척 체크

#### 트러블 슈팅

개인 -> 팀 slack -> 기술멘토와 질의응답 slack

#### 말하고 싶은 거 있으면 자유롭게 이야기 하기

개개인 : Slack DM / 전화  
팀 문제 : Slack 팀 채팅방 / 정기회의 안건으로 제시



## 위험요소 및 극복제도 (Git)

### 3.1. 위험요소

**01** 푸시한 것들이 꼬여서  
백업에 시간을 쏟아야 하는 상황

**02** 일관성 없는 커밋 메세지

- 팀원이 커밋 히스토리 파악하는데 시간을 줄여야함
- 원하는 작업 내용을 찾기 쉬워야함

### 3.2. 극복제도

**리소스에 맞는 브랜칭 전략 세우기**

- Git flow 전략을 참고한 브랜칭 전략
- feature 브랜치 : [개발파트/기능]

**히스토리 파악이 가능한 커밋 메세지 작성**

- 널리 통용되고 있는 "Chris Beams가 작성한 커밋 메세지 작성 규칙" 사용.



# 감사합니다!



팀장 박신형

팀원 김다빈

팀원 박형우

팀원 이유성