

sMilestone's PMP



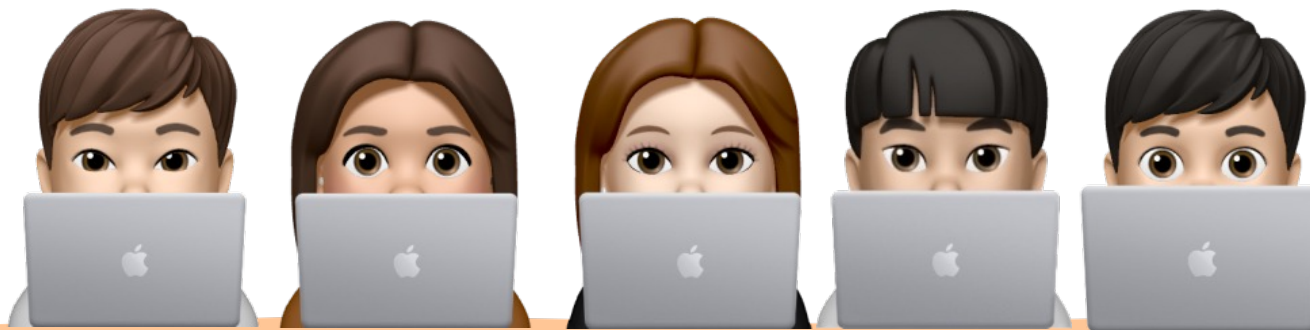
목차

- ✓ Goal Setting(Team/개인)
- ✓ How To Work
- ✓ What Is Our Platform ?
- ✓ WorkFlow
- ✓ 개발계획(Team/개인)





Goal Setting



- 앱/웹 친화적인 서비스
- 모든 플랫폼에서 통일된 UI를 제공할 수 있다.
- 다양한 플랫폼에서 개발하는 개인의 목표를 이룰 수 있다.

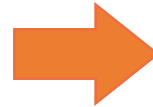


Goal Setting



AS-IS

- 반응형으로 보이는 화면에만 집중하는 개발
- 정리되지 않은 디렉토리 구조로 개발을 진행하는 습관
- 특정한 패턴이 없이 혼란스러운 코드작성



TO-BE

- React 커스텀 혹, 컴포넌트화로 재사용성 있는 코드 작성이 가능
- 유형 별로 파일을 분류해 가독성 높은 디렉토리 관리가 가능



- 사용자 관점에서 문제를 해결하도록 UI/UX 실력 향상하기
- Atomic Design Pattern으로 일관성 있는 코드 작성하기
- webRTC 구조 이해하여 백엔드와 원활한 연결하기



이다혜

Goal Setting



AS-IS

- 공식 문서를 잘 참고하지 않고 개발하는 경우가 많음
- Testable 한 코드를 개발해 본 경험이 없음



TO-BE

- GitHub Swift Architecture 문서를 파악하여 코드 스타일 맞춰가며 개발
- Apple API Design Guidelines 공식 문서를 참고하여 개발
- Apple의 디자인 철학 Human Interface Guidelines 를 공부한 후 그에 맞춰서 UI 설계
- WWDC17에서 발표된 Engineering for Testability 내용을 보고 앱의 크기나 복잡성이 커져도 대응 가능한 확정성이 있는 Test코드를 작성
- AAA(Arange-Art-Assert) 패턴에 맞춰 Test 코드 작성



- GitHub Swift Architecture 문서를 파악하여 코드 스타일 맞춰가며 개발
- Apple API Design Guidelines 공식 문서를 참고하여 개발
- Apple 의 디자인 철학 Human Interface Guidelines 을 공부 후 그에 맞춰서 UI 설계
- Testable 한 코드 개발을 위해 노력



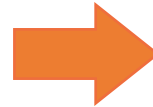
김경욱

Goal Setting



AS-IS

- 프레임워크의 기능을 제대로 이해하지 않고 의존하는 습관
- 실시간 통신 프로젝트에 대한 경험 부족
- 성능 테스트에 대한 경험 부족



TO-BE

- Docs를 이해하고 작동 원리를 바탕으로 포스팅 작성
- 프레임 워크, 오픈 소스 선택에 이유를 명확히 남기기
- Custom을 통한 기능 응용
- 1:1 (1:다) 실시간 채팅을 구현해보기 (10마디 이상)
- 팀과 함께 세운 성능 테스트 목표를 달성하기



- 실시간 통신의 과정을 숙지 후 실시간 채팅 구현해보기
- 효율적이고 이유 있는 시스템 설계 하기
- 성능 테스트 경험 및 목표 달성하기



김성운

Goal Setting



AS-IS

- 모바일 디자인 패턴을 상관하지 않고 개발을 진행하는 경우가 많음
- 모르는 부분을 구글링하고 해결하기 급급하고 그것을 정리하여 나의 것으로 만들지 않는 경우가 대부분
- 백엔드와의 연결을 해 본 경험이 별로 없음
- UI/UX 디자인이 많이 부족함



TO-BE

- MVVM 패턴에 대해 유튜브, 구글링을 통해 공부하고 MVVM 패턴을 통해 만들어진 예제를 분석한 후 실제 프로젝트에 MVVM 패턴을 적용
- 프로젝트 기간 동안 개발 과정에서 막힌 부분을 블로그에 기록 후, 해결 과정 및 지식을 정리한다. 또한, 모든 포스트에는 성찰을 통한 자기 성장의 내용이 담기도록 한다.
- 1월 동안 Retrofit, Volley를 통한 http 통신 방법에 대해 숙지하고 JSON, 이미지, 음성 데이터 등을 주고 받는 코드를 작성하고 블로그에 정리
- 프로젝트 기간 동안 안드로이드 디자인 가이드라인에서 제시한 디자인을 적용하여 UI/UX 디자인 실력 향상



- MVVM 패턴을 이해하고 경험해보자
- 개발과정에서 막힌 부분들과 해결한 과정과 지식 그리고 성찰 부분 정리
- 백엔드와 연결 라이브러리 완전 숙지
- UI/UX 디자인 실력 향상



김근범

Goal Setting



AS-IS

- 새롭게 배운 지식을 바탕으로 코드 작성 시, 정확한 이해를 바탕으로 코드를 작성하지 않아 코드를 통째로 폐기하는 일이 잦았다. 이로 인해 개발 일정에 차질이 생기는 일이 굉장히 잦았음
- 실시간 플랫폼(webRTC)에 관심이 많음
- 의미 없는 commit message만 작성



TO-BE

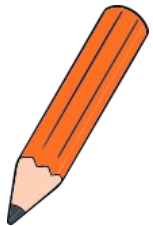
- 시스템을 정확히 이해하고 개발하여 코드를 다시 작성하지 않는 개발자
- Git을 활용하여 협업 상황에서 Git 숙지



- webRTC를 통해 개인 방송 플랫폼 개발
- 오픈소스 미디어 서버를 이용하여 미디어(오디오, 비디오) 통신
- Web Socket을 통해 1주 안에 채팅 서버 개발
- 필요한(인증, 채팅 내역 등) Web API Spring Boot를 통해 개발
- 팀원과의 다양한 갈등 해소를 통해 갈등 해소 방법 터득
- 매일 smilegate에서 배운 개발 경험 문서화와 그를 통한 회고, 성찰하기



남정진



How To Work



이다혜

사용자 관점에서 문제를 해결하도록 UI/UX 실력 향상하기

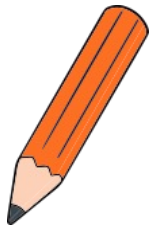
- UI/UX 레퍼런스 사이트 참고하여 Figma로 따라해보면 감을 익히기

Atomic Design Pattern으로 일관성 있는 코드 작성하기

- 공식 문서를 통해 Atoms, Molecules, Organisms, Template의 전반적인 내용을 정리하여 개인 블로그에 포스팅하며 이해하기
- home 화면, blog 화면, 영상통화 화면, chatting 화면 등 모든 화면에 보이는 요소를 수집 후 그 요소들을 가장 작은 단위로 쪼개기

webRTC 구조를 이해하여 백엔드와 원활한 연결하기

- 공식 문서를 읽어보고 내용을 정리하여 포스팅하여 이해하기
- FE와 BE를 연동하여 구현한 사례를 참고하여 방법 숙지하기



How To Work



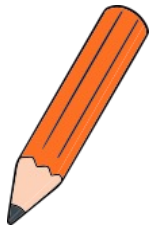
김경욱

Apple의 공식 문서를 참고하여 개발하기

- Swift Architecture 문서를 파악하여 코드 스타일 맞춰가며 개발
- Apple API Design Guidelines 공식 문서를 참고하여 개발
- Apple의 디자인 철학 Human Interface Guidelines 를 공부한 후 그에 맞춰서 UI 설계

Testable한 코드 개발을 위해 노력

- WWDC17에서 발표된 Engineering for Testability 내용을 보고 앱의 크기나 복잡성이 커져도 대응 가능한 확정성이 있는 Test코드를 작성
- AAA(Arange-Art-Assert) 패턴에 맞춰 Test 코드 작성
- 입력을 제어할 수 있고, 출력을 검사할 수 있고, 코드의 동작에 영향을 줄 수 있는 내부 상태에 의존하지 않는 특성에 맞추어 개발



How To Work



김성운

실시간 통신을 완전 숙지 후 실시간 채팅 구현해보기

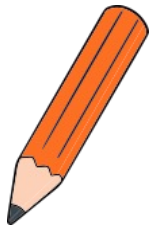
- STOMP, SockJS 공식 Docs를 읽고 파악하여 실시간 채팅 구현하기
- FE와의 연결을 위한 SockJS 학습 후 개발해보기

효율적이고 이유 있는 시스템 설계 하기

- 필요한 기능을 정의한 뒤, 걸맞는 아키텍처 선택 이유를 명확하게 정리하기
- 다양한 툴 안에서 내가 이해한 원리를 블로그 포스팅으로 정리하기

성능 테스트 경험 및 개선을 통해 목표 달성하기

- 제공되는 기능을 Custom하여 성능 개선하기
- 메시지 큐 시스템 도입해보기 (Optional)



How To Work



김근범

MVVM 패턴을 이해하고 경험

- 기존에 구현되어 있는 MVVM 패턴을 많이 보고 분석
- 개발 전에 MVVM 패턴으로 아키텍처 설계
- 뷰 영역 / 뷰 모델 영역 / 모델 영역 확실히 나누기

백엔드와 연결 연습

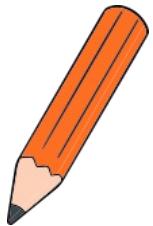
- Retrofit과 Volley를 통한 http 연결 방법에 대해 확실히 학습하고 정리
- 이미지, 동영상, 음성, 데이터에 대한 데이터 전달 예제 코드 작성 연습

개발 과정에서 막힌 부분들과 해결한 과정과 지식, 성찰 부분 정리

- 개발 과정에서 막힌 부분들을 모두 기록하기
- 막힌 부분들을 해결한 방법에 대해 기록하고 그에 대한 지식들도 정리
- 모든 정리에는 성찰에 대한 내용을 반드시 작성

UI/UX 디자인 실력 향상

- 잘 구현된 UI, UX 예제들을 보고 공부해보기
- 프로젝트 기간 동안 안드로이드 디자인 가이드라인에서 제시한 디자인을 적용
- UI, UX 구현 시 필요한 내용들은 유튜브나 구글을 통해 지식 쌓기



How To Work



남정진

Github의 코드를 클론하여 WebRTC 실시간 화상 플랫폼 구현
(Mesh Topology, Mediasoup)

클론을 통한 Mediasoup 이해를 바탕으로 Room 기반의 실시간
화상 플랫폼 구현

막히거나 오류가 나는 부분은 기록하며 왜 발생했는지 기록하며
정확히 문제 해결

위의 과정을 통해 시스템을 잘못 이해하여 코드를 통째로 폐기해야 하는
상황 방지

Git log를 정리하고 정확한 메시지 작성 후에 push



What Is Our Platform?

기술 공유 플랫폼 (당근마켓 + 디스코드)

💡 사용자 간에 수학, 과학 등의 여러가지 기술들을 사고 팔 수 있는 플랫폼

• 웹/앱 공통

- ✓ 로그인/가입
- ✓ 이메일로 계정 찾기
- ✓ 홈 (당근마켓과 유사)
- ✓ 검색바
- ✓ 판매화면 (글 제목, 가격 입력, 게시글 내용 작성, 사진 첨부)
- ✓ 채팅 (구매자-판매자)

• 웹

- ✓ 1:N 채팅 대화
- ✓ 1:N 영상통화
- ✓ 1:N 화면공유 (Optional)

• 앱

- ✓ 1:N 채팅 대화
- ✓ 1:N 전화



sMilestone's WorkFlow

정기 회의

- **매주 화요일 1번 정기 회의**
(일시 : 23시-24시 , 장소 : Zep)
- **회의록 순서 정해 돌아가면서 작성**
(김성윤 - 김경욱 - 김근범 - 이다혜 - 남정진)
- **협업툴**
(Slack, Zep, Git, Notion, Kakao)



Ground Rules

- 공지사항 및 팀원 메신저에 이모지를 적극적으로 활용하여 읽음을 어필한다.
- 회의 시간에 회의 안건에 대해 최소 1가지 이상의 의견을 제시한다.
- 서로의 속도 차이를 이해한다.



역할

- 이다혜 : 팀장
- 김경욱 : 부 팀장
- 김성윤 : 서기 (문서 작업)
- 남정진 : 기술 팀장
- 김근범 : 스크럼 매니저



형상 관리

- 기능 별 git branch



sMilestone's 개발계획



Weekly Scrum

- 이번 주 각자 진행 상황 공유
- 다음 주 회의 때 의논할 사항
- 현재 가장 시급하게 해결해야 하는 안건
- 다음 주까지 개인 별 진행 예정
- 우선 순위 별 할 일 정리
- 질문 사항
- 주간 회의를 통해 개발 현황 기록



<Notion 작성>



Weekly Scrum 1(1/5~1/10)

이번 주 개인별 진행 상황 공유

개인별 진행한 작업 정리

☐ 완료	Who?	분야	Aa 내용
<input type="checkbox"/>	③ sienna	BE	제목 없음
<input type="checkbox"/>	👤 JJ NAM	BE	제목 없음
<input type="checkbox"/>	👤 다혜 이	FE	제목 없음
<input type="checkbox"/>	👤 Kim_KU	iOS	제목 없음
<input type="checkbox"/>	👤 근범 김	Android	제목 없음

다음 주까지 개인별 진행 예정인 작업

개인별 진행 예정 작업 정리

Who?	태그	Aa 내용
③ sienna	BE	제목 없음
👤 JJ NAM	BE	제목 없음
👤 다혜 이	FE	제목 없음
👤 Kim_KU	iOS	제목 없음
👤 근범 김	Android	제목 없음

다음 주 회의 때 의논할 사항

- 자유롭게 추가해주세요



sMilestone's 개발계획



MILESTONE

- 인증 기능
- REST 기능
- 실시간 기능

🔥 집중 개발 기간 : 최종 서비스의 기능을 8개로 나누어 우선 순위에 따라 각 주마다 해결

월	1월				2월			
주차	1주차	2주차	3주차	4주차	1주차	2주차	3주차	4주차
Smilegate 계획	PMP 리뷰 아키텍처 설계	PMP 리뷰 아키텍처 설계	중간 공유회	집중 개발 기간 시작		코드리뷰	코드리뷰 및 성능 테스트	최종 발표 준비
research	아키텍처 설계 및 정의서 작성 개발 지식 쌓기							
develop		각 모듈 별 기능 개발 및 연동	각 모듈 별 기능 개발 및 연동	각 모듈 별 기능 개발 및 연동	각 모듈 별 기능 개발 및 연동	각 모듈 별 기능 개발 및 연동 코드 리뷰 바탕으로 리팩토링	코드리뷰 바탕으로 리팩토링	
testing			통합 테스트		통합, 성능 테스트 및 개선(1차)		통합, 성능 테스트 및 개선(2차)	
docs								발표 준비



IOS / Android 개발 계획 (김경욱 / 김근범)

사전 분석 (~1/9)

- 어플리케이션 레이아웃(채팅 서버, 설정, 친구 관리 등) 분석 (당근, 디스코드, 인프런 등 참조)
- 필요한 API 및 라이브러리 공부

UI 구현 (~1/14)

- Figma를 통한 UI 설계

로그인 기능 구현 (~1/18)

- 로그인 기능 구현 및 인증 서버와 연결
- 로그인 방법(구글, 카카오 등) 고민 필요

홈 화면 및 검색 바 구현 (~1/21)

- 리사이클러 뷰로 판매자들이 올린 글들 리스트 제공
- 검색 바는 UI만 생성

검색 기능 구현 (~1/24)

- 데이터베이스에서 검색어로 검색할 수 있게 구현

기술 공유 화면 및 판매 글 등록 기능 구현 (~1/30)

- 공유 글에 대한 CRUD 구현

채팅 기능 구현 (~2/5)

- 서버와 연결하기 위한 라이브러리 결정(retrofit, volley 그 외)
- 채팅 시 보낼 수 있는 데이터 타입 결정(글, 이미지, 음성 파일 등)
- 데이터 타입에 따른 채팅 서버와 연결 로직 구성

음성 채팅 기능 구현 (~2/12)

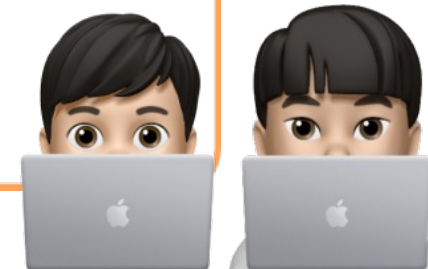
- 아마 WebRTC 기능을 쓰지 않을까?

설정 및 기술 공유자 정보 구현 (~2/14)

- 비밀번호 변경, 프로필 변경 등
- 판매자 프로필 및 기타 정보 제공

피드백 및 성능 테스트 후 리팩토링 (~2/17)

- 코드 리뷰 및 성능 테스트 진행
- 미흡한 부분 리팩토링





BE 개발 계획 (김성윤 / 남정진)

사전 분석 (~1/9)

- Docs 훑아보기, Git 에서 토이 프로젝트 탐색

시스템 설계 (~1/11)

- 프레임 워크 비교 및 선택
- 기능 정의서, ERD, 기능 명세서 작성

회원가입 (~1/14)

- 기본 REST 로직으로 구현

로그인 기능 구현 (~1/18)

- OAuth2를 사용하여 구현

채팅 기능 구현, 캐싱 (optional) (~2/5)

- STOMP, SockJS 등 프레임 워크 탐색 후 구현

피드백 및 성능 테스트 후 리팩토링 (~2/17)



개발 환경 구축, UI/UX 설계 (~1/14)

Room CRUD, 인가, 캐싱 (~1/21)

화상 서버 구현 (~1/28)

화상 기능 구현 (~2/04)

SNS 기능, 검색 구현 (~2/11)

피드백 및 성능 테스트 후 리팩토링 (~2/17)





FE 개발 계획 (이다혜)

개발 환경 세팅 및 화면 구성 분석 (~1/9)

- 개발 환경 구축하기
- 당근마켓, 디스코드 등 웹 화면 분석하여 컴포넌트 분리 작업

UI 구현 (~1/14)

- Figma를 이용하여 UI 설계

Atom 작성 (~1/18)

- 화면 구성 분석을 토대로 Atom 작성하기 (props로 전달받아 사용할 수 있도록 구현)
- 회원가입, 로그인 form 작성

Molecules, Organism (~1/31)

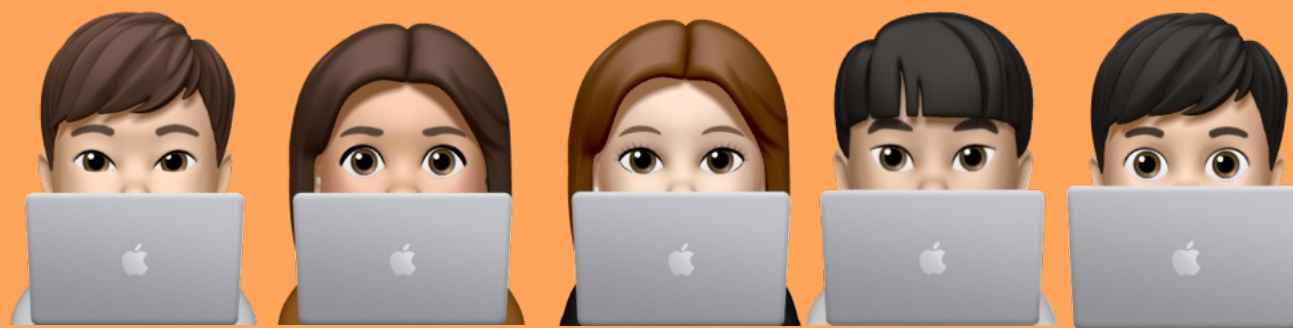
- Atom을 기반으로 컴포넌트화 하기

BE와 연동 (~2/12)

- 글 게시 화면, 영상 통화, 1:1 (1:n) 채팅 등 BE와 연동

피드백 및 성능 테스트 후 리팩토링 (~2/17)





Thank you

