VILLAGE

공간 대여 앱

Final project

4조 박인우 김정욱 김호현 이인화

목차

01 프로젝트 소개

-프로젝트 소개

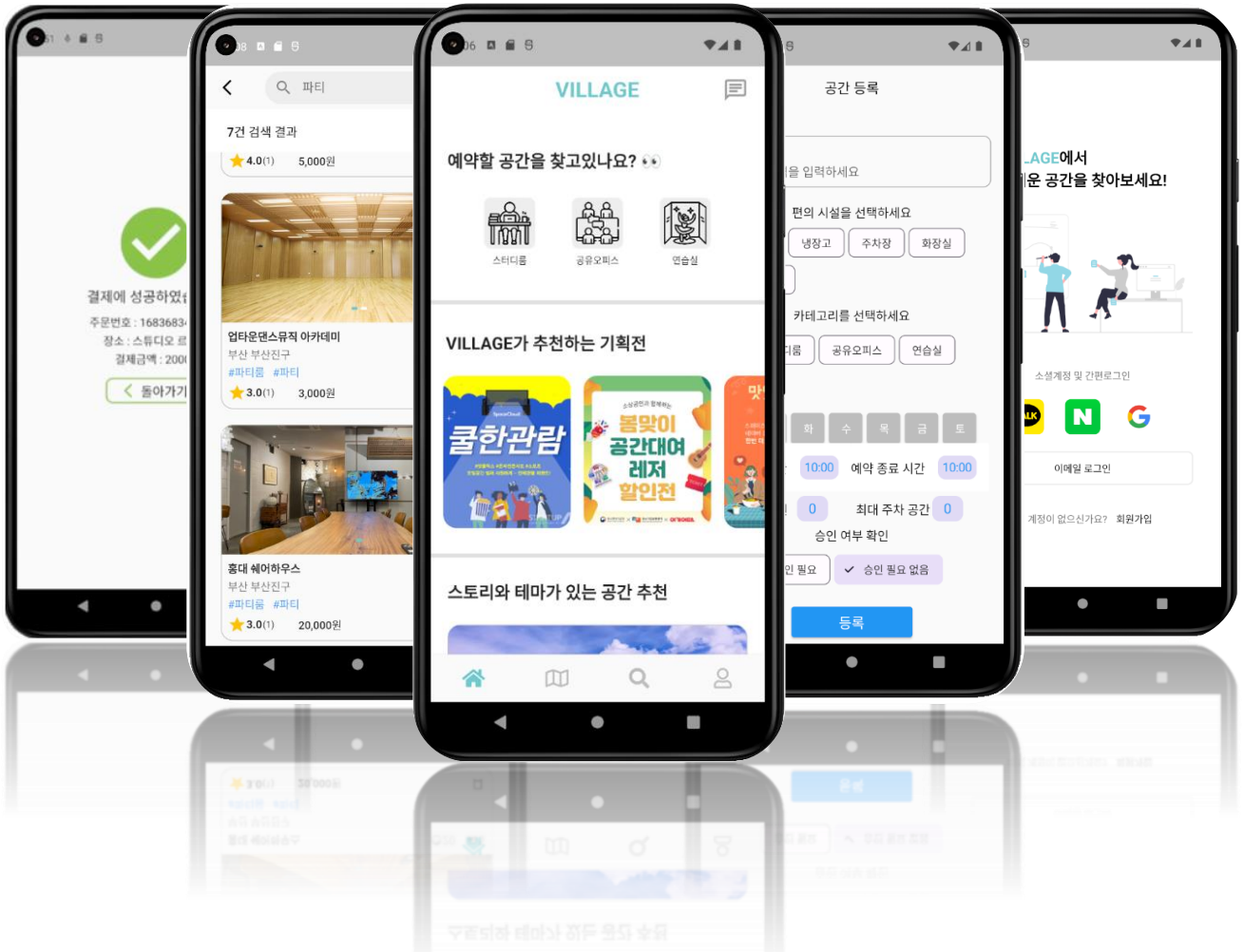
-프로젝트 목표

-사용자 시나리오

-개발환경

-협업 과정

01 프로젝트 소개



레퍼런스 앱

SpaceCloud. 스페이스 클라우드

Billyo 빌리오

만나다

슈잉

VILLAGE 앱은
참여기업의 중개 플랫폼 웹/앱 개발 요구에 따라
개발된 공간 대여 중개플랫폼으로
사용자들이 자신의 공간을
다른 사람들에게 임대하거나,
필요한 공간을 찾을 수 있도록 연결해주는
플랫폼입니다.

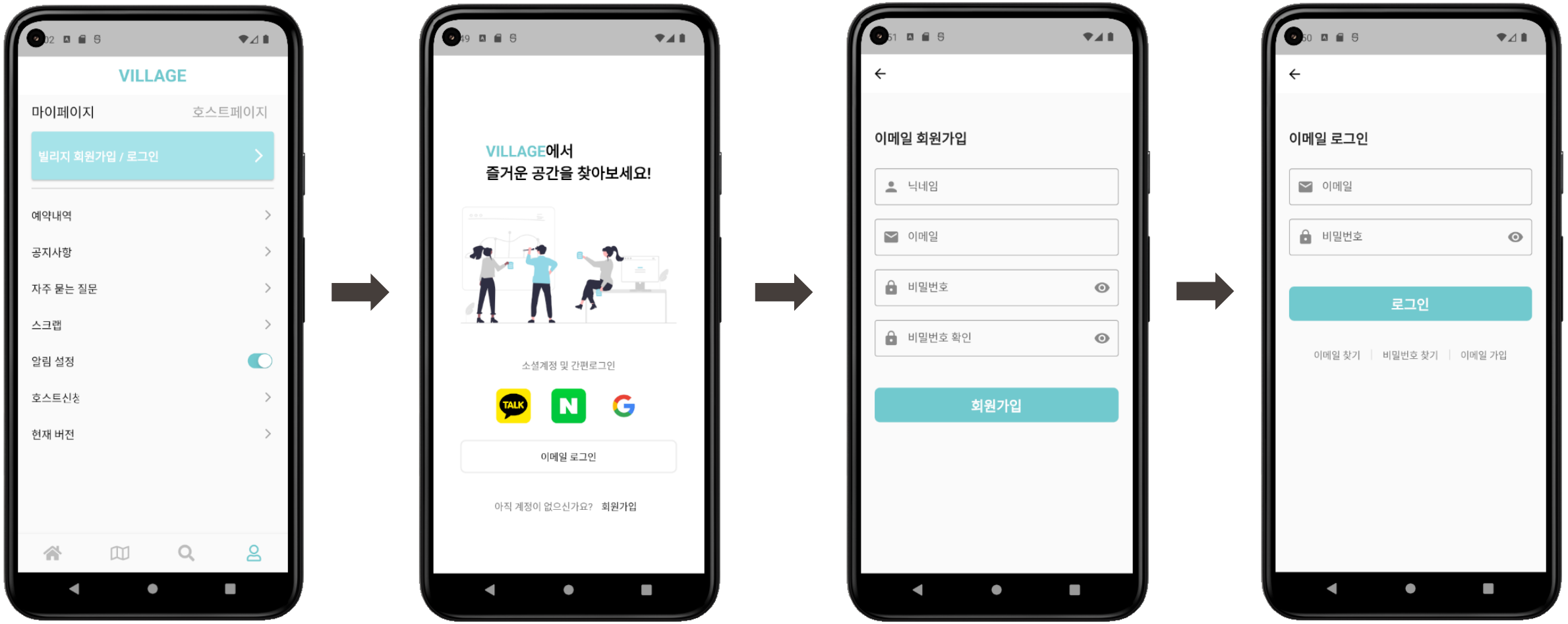
01 프로젝트 목표

[목표]

- 플랫폼 기능 및 UI/UX 설계
- 서버 설계
- 데이터베이스 선정
- 데이터베이스 설계 및 구축
- 서버와 데이터베이스 인터페이스 구축
- 플랫폼 운영, 관리 시스템 구현
- 상품 검색 기능 구현
- 입점 가맹점의 관리기능 구현
- 데이터 통합관리 모듈 설계
- 시각화 모델 개발
- 실시간 온라인 예약 및 결제시스템 구현
- 테스트 및 배포

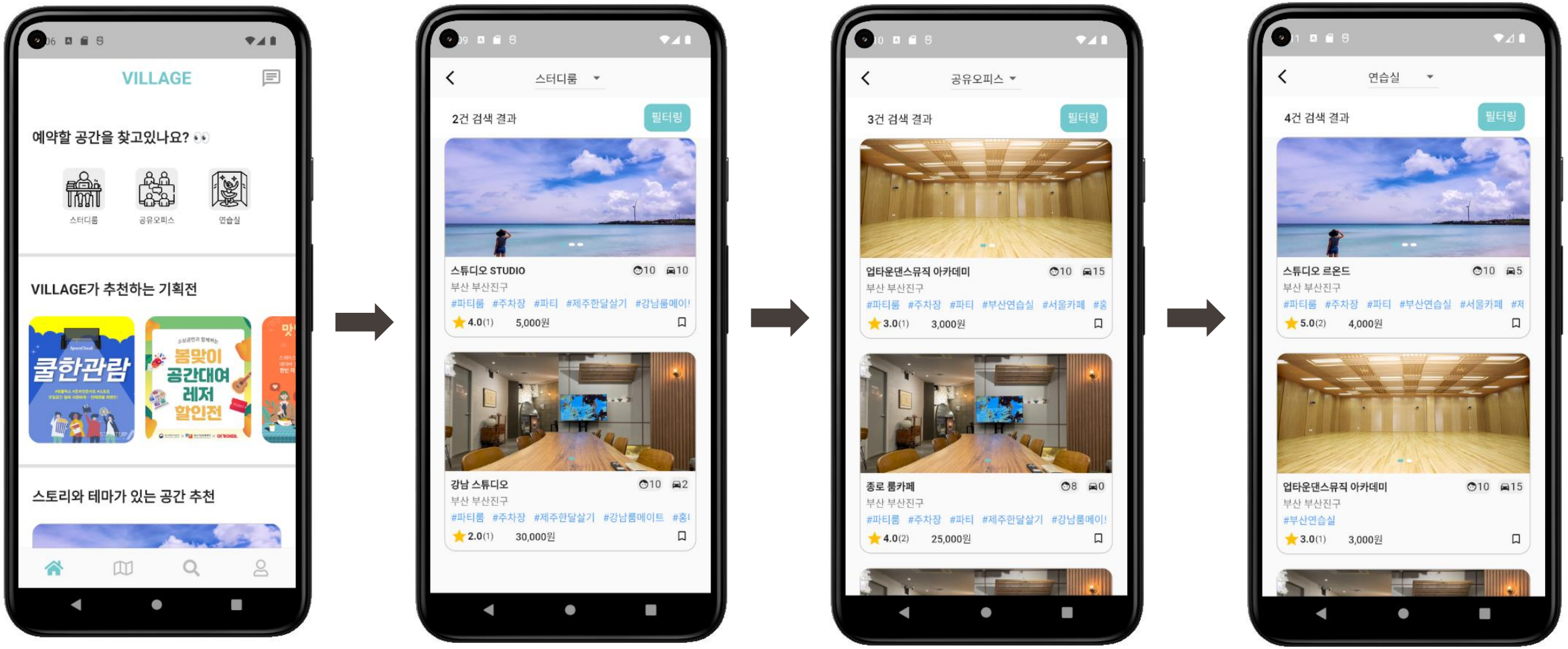
01 사용자 시나리오

회원가입 및 로그인



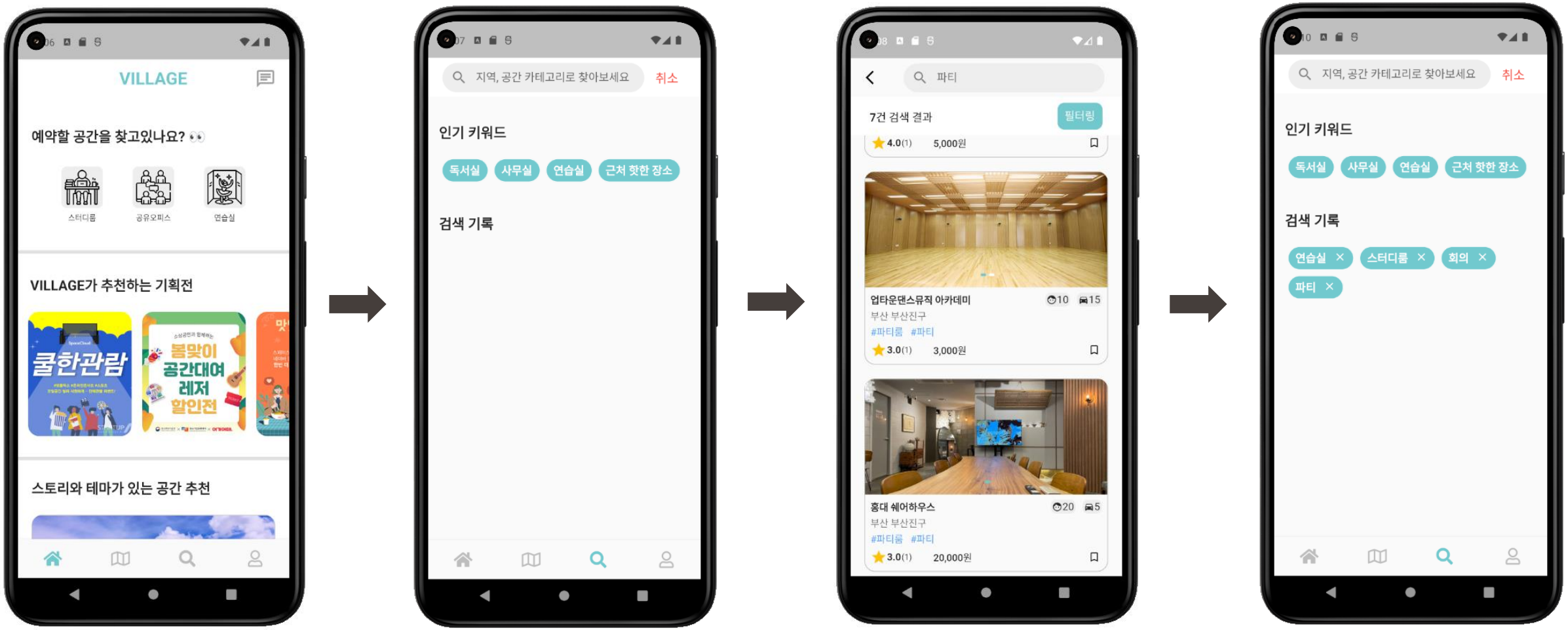
01 사용자 시나리오

카테고리 별 목록(스터디룸, 공유 오피스, 연습실)



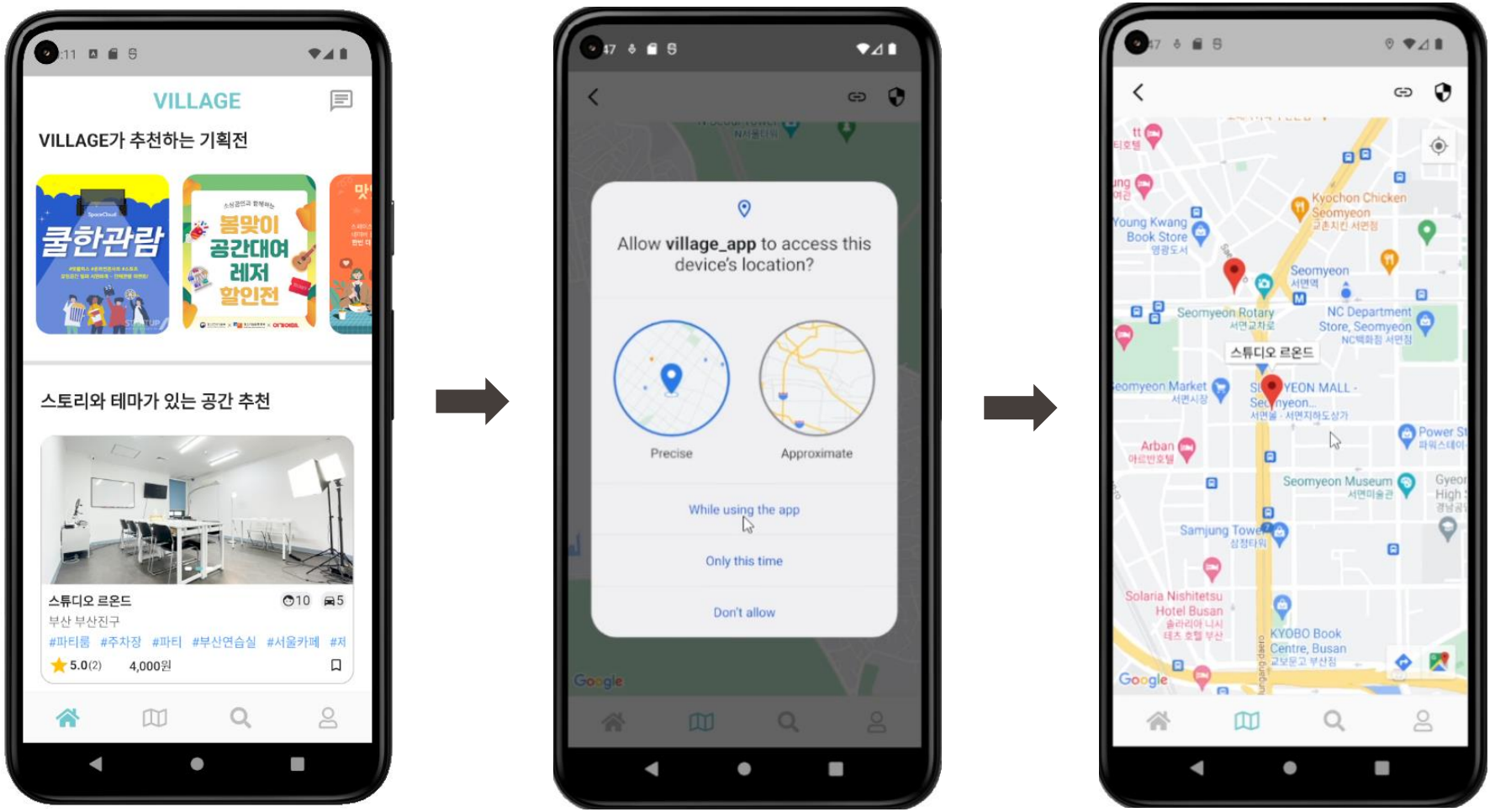
01 사용자 시나리오

검색하기



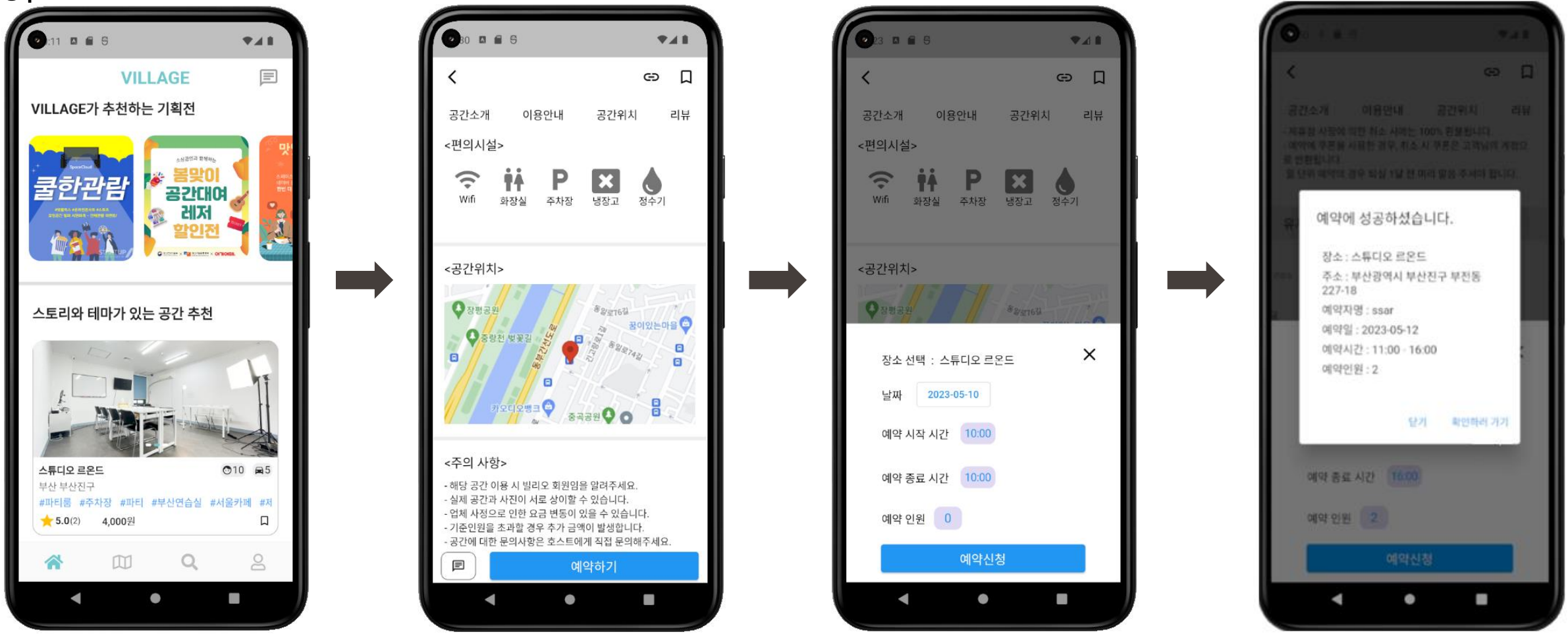
01 사용자 시나리오

지도



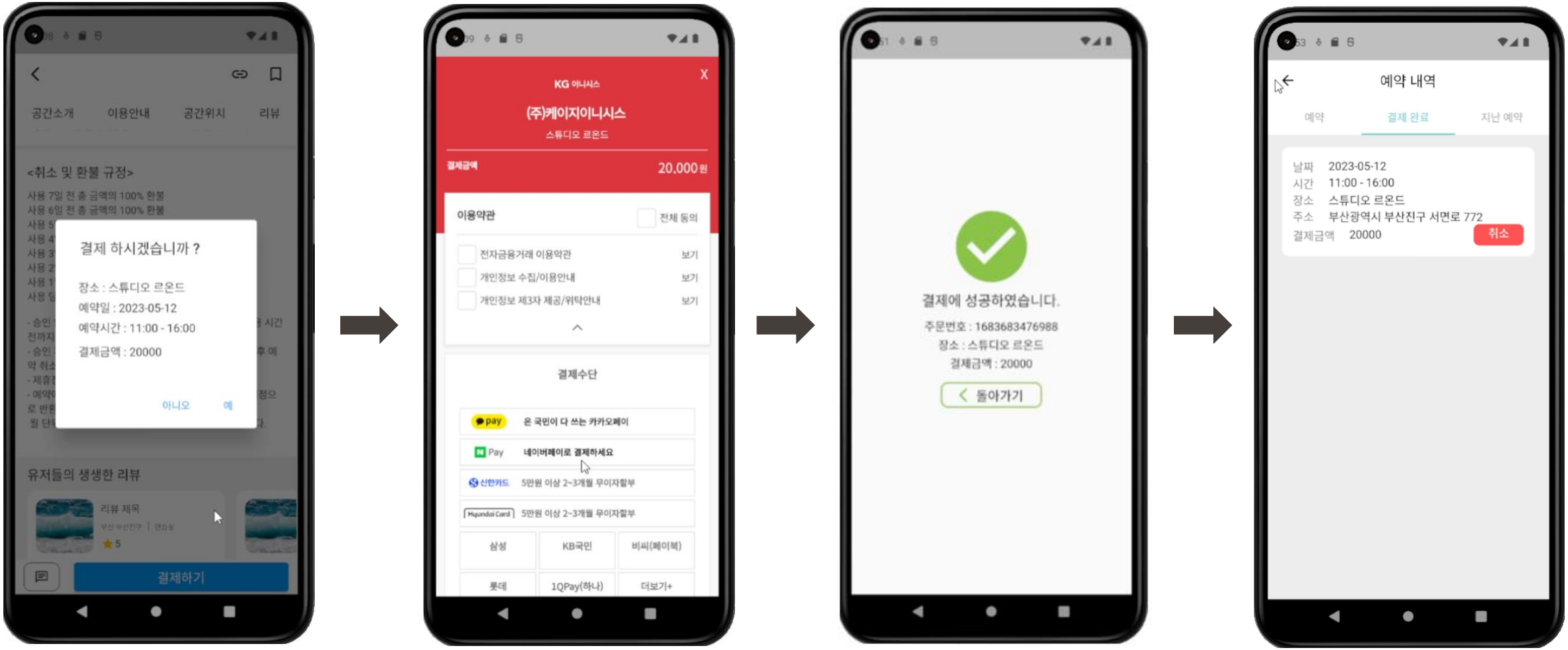
01 사용자 시나리오

공간 상세보기 및 예약



01 사용자 시나리오

결제

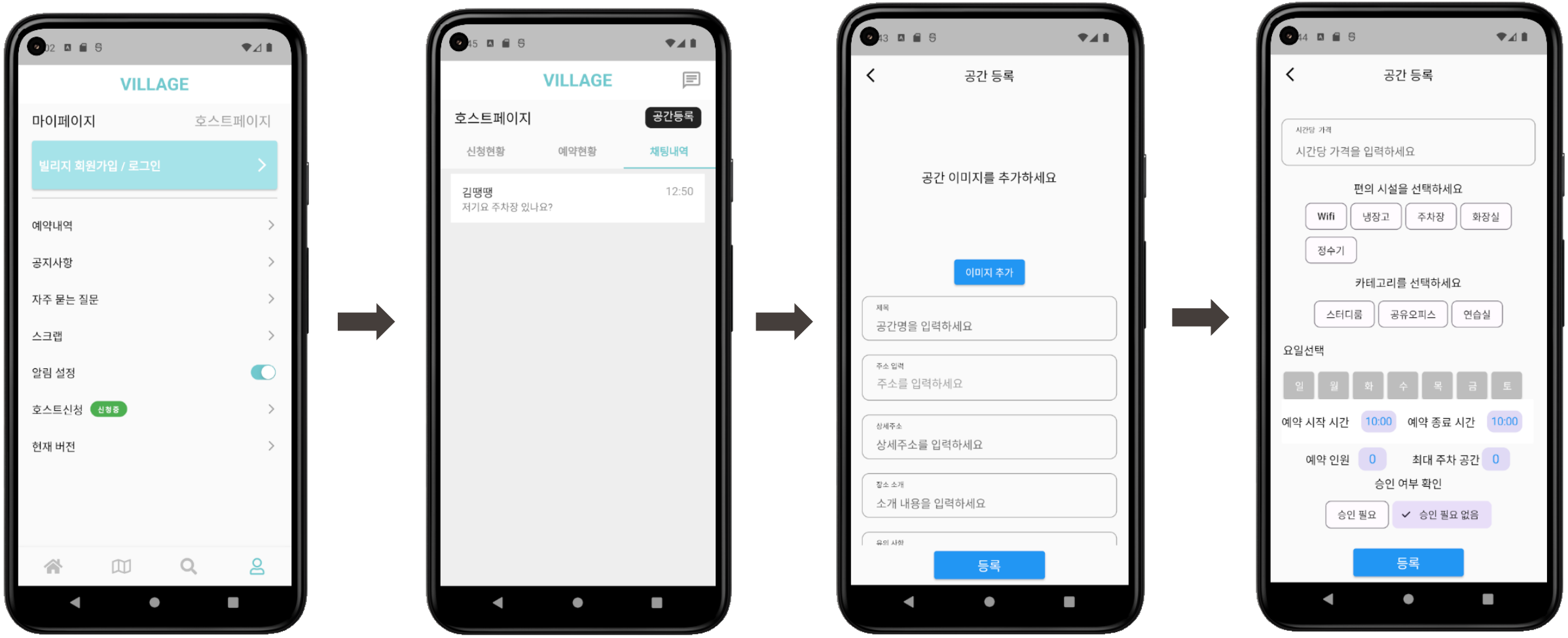


01 호스트 시나리오

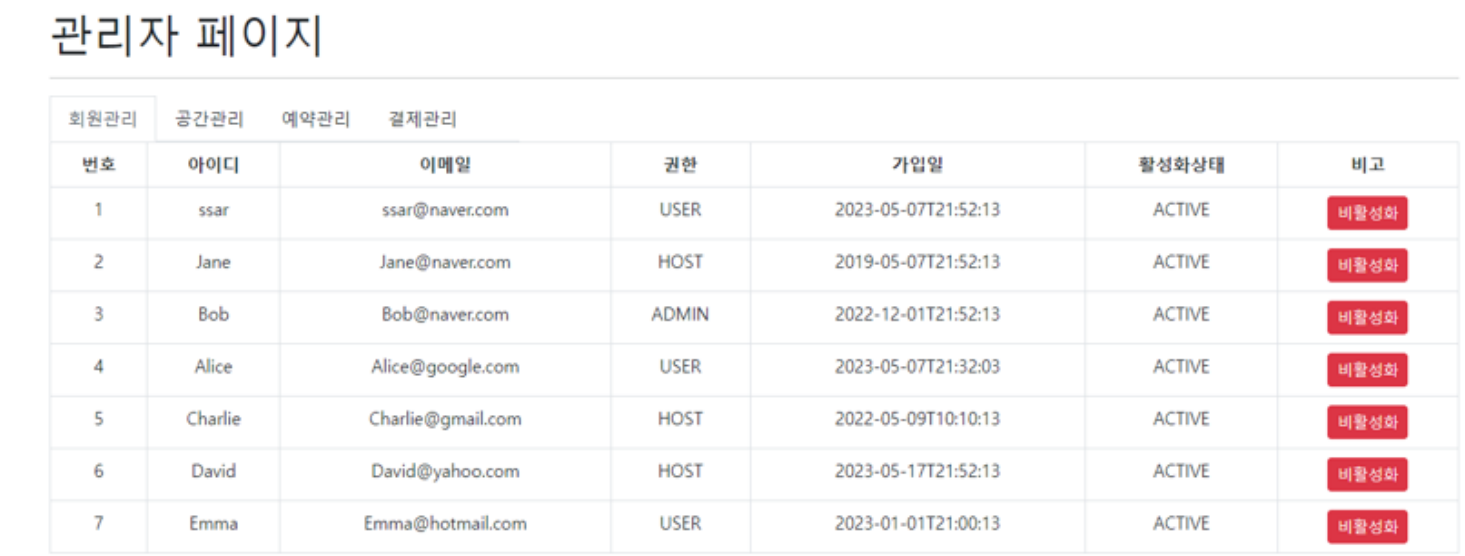
호스트 신청

01 호스트 시나리오

호스트 공간 등록



01 관리자 웹 메인



관리자 기능

-회원 관리
-공간 관리
-예약 관리
-결제 관리

01 관리자 웹 호스트 신청 관리

01 개발 환경

개발 툴

Visual Studio Code

IJ

android studio

언어

Java

Dart

DB

H2

MariaDB

프레임워크

Flutter

spring boot

사용 기술

riverpod

spring SECURITY

aws

JPA
Java Persistence API
HIBERNATE

협업 툴

Figma

N

POSTMAN

git

GitHub

01 협업 과정 - git & github branch 전략

Rebase 전략

-초기에는 3 way-merge 전략 사용으로 팀장이 충돌을 처리했다.

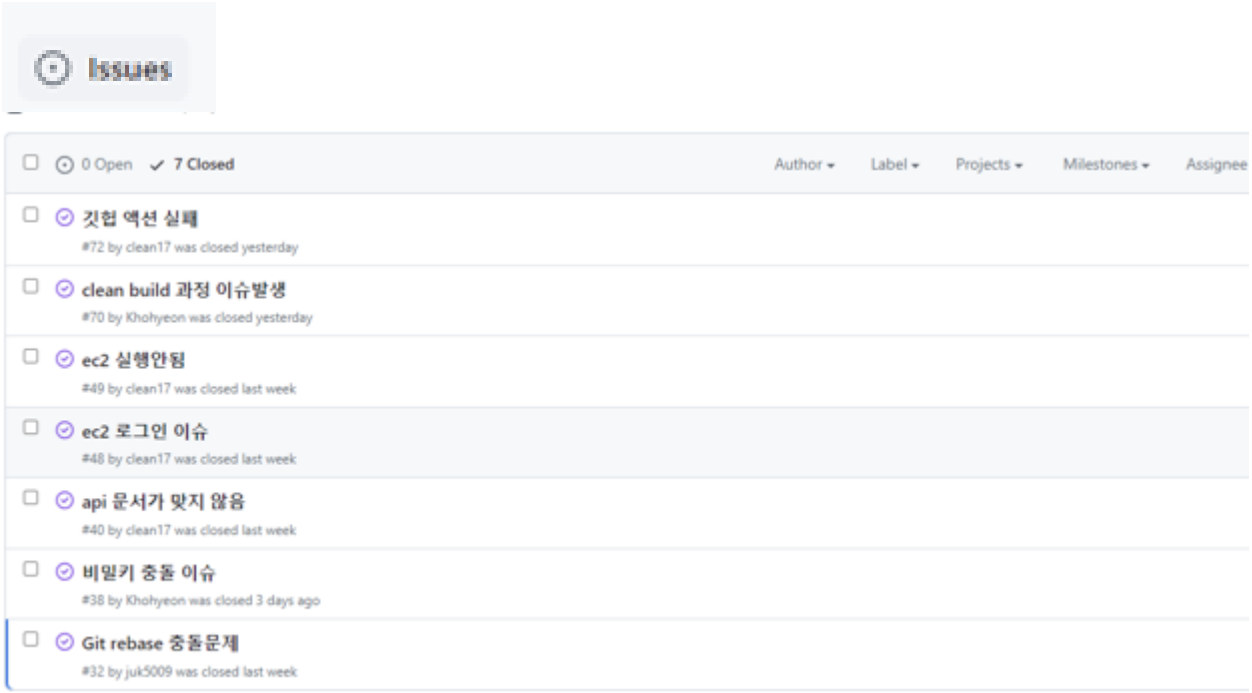
-Rebase전략으로 변경하여 개인이 충돌을 해결하여 pr요청 하는 것으로 방법으로 팀장의 부담을 줄였다.

- commit 히스토리가 깔끔해져서 추적하기 쉬워졌다.

01 협업 과정 - git & github branch 전략

git issues

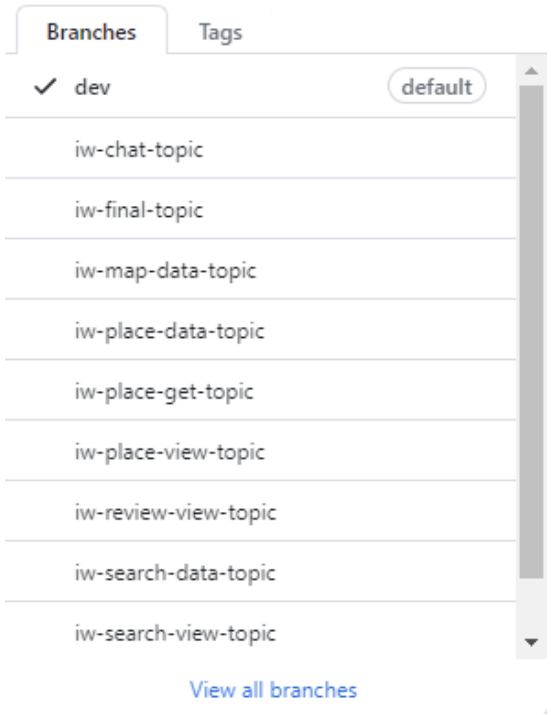
- 프로젝트 기간 중 발생한 오류를 기록해 지난 오류 로그를 한눈에 파악하기 좋았다.
- 여유로운 팀원이 이슈를 확인하고 해결했다.



01 협업 과정 - git & github branch 전략

[공통] - 본인 이름 이니셜을 붙여 구분

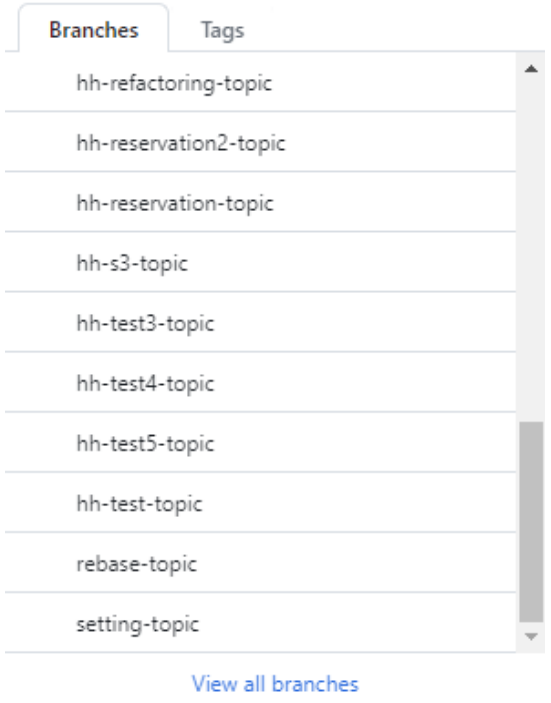
[프론트엔드]



화면 만들 때 view-topic

바인딩할 때 data-topic

[백엔드]



기능 별 - topic

Test 할 때는 test - topic

01 협업 과정 – Notion 스케줄 관리

에픽 캘린더 | 작업 | 엔지니어별 작업 | 상태별 작업 | 타임라인 +

clean17 7

- 센트리로 로그 관리
- 공간 데이터 조회
- 초기 세팅
- 주소 검색시 위도 경도 통신
- 공간 등록 데이터 프로바이더 저장
- ec2 배포
- 플러터 화면 뷰

+ 새로 만들기

김호현 4

- 결제 기능 구현
  우선순위1
- 공간 등록 구현
- 예약 기능 구현
- 예약 신청 승인 여부 처리

+ 새로 만들기

이인화 6

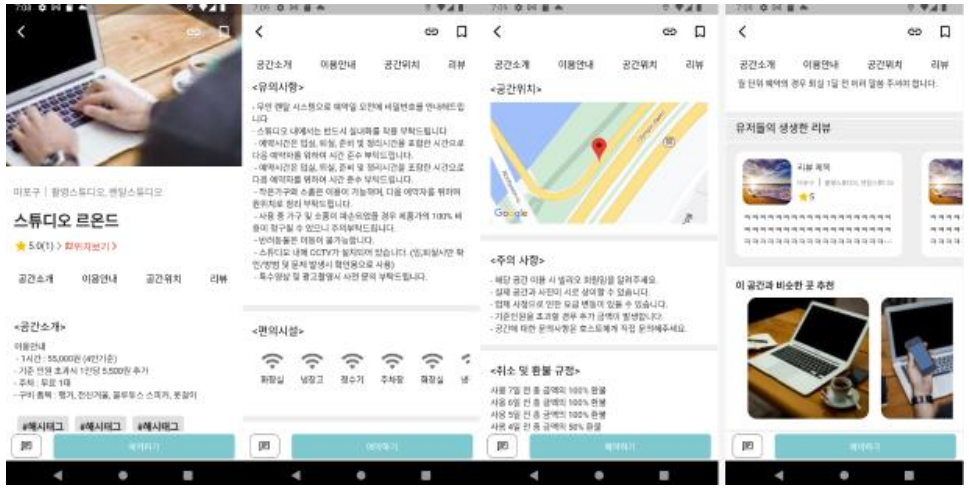
- flutter 화면
- 회원가입 연결
- 로그인 연결
- 호스트 신청 연결
- 검색 연결
- 더미 사진 필요

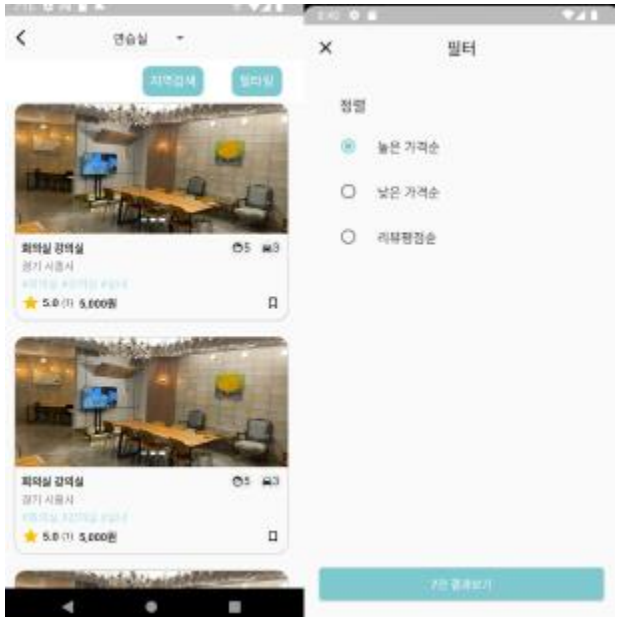
+ 새로 만들기

JeongUk 6

- 로그인 구현
  우선순위1
- 회원가입 중복 처리
  우선순위1
- Exception 처리 수정
  우선순위1
- 더미 패스워드 해시로 입력
  우선순위2
- 검색 리팩토링
- 필터링 (정렬)기능

01 협업 과정 – 피그마





피그마로 화면을 공유하여 서로 필요한 데이터 및 기능을 회의했습니다.

02 프론트엔드

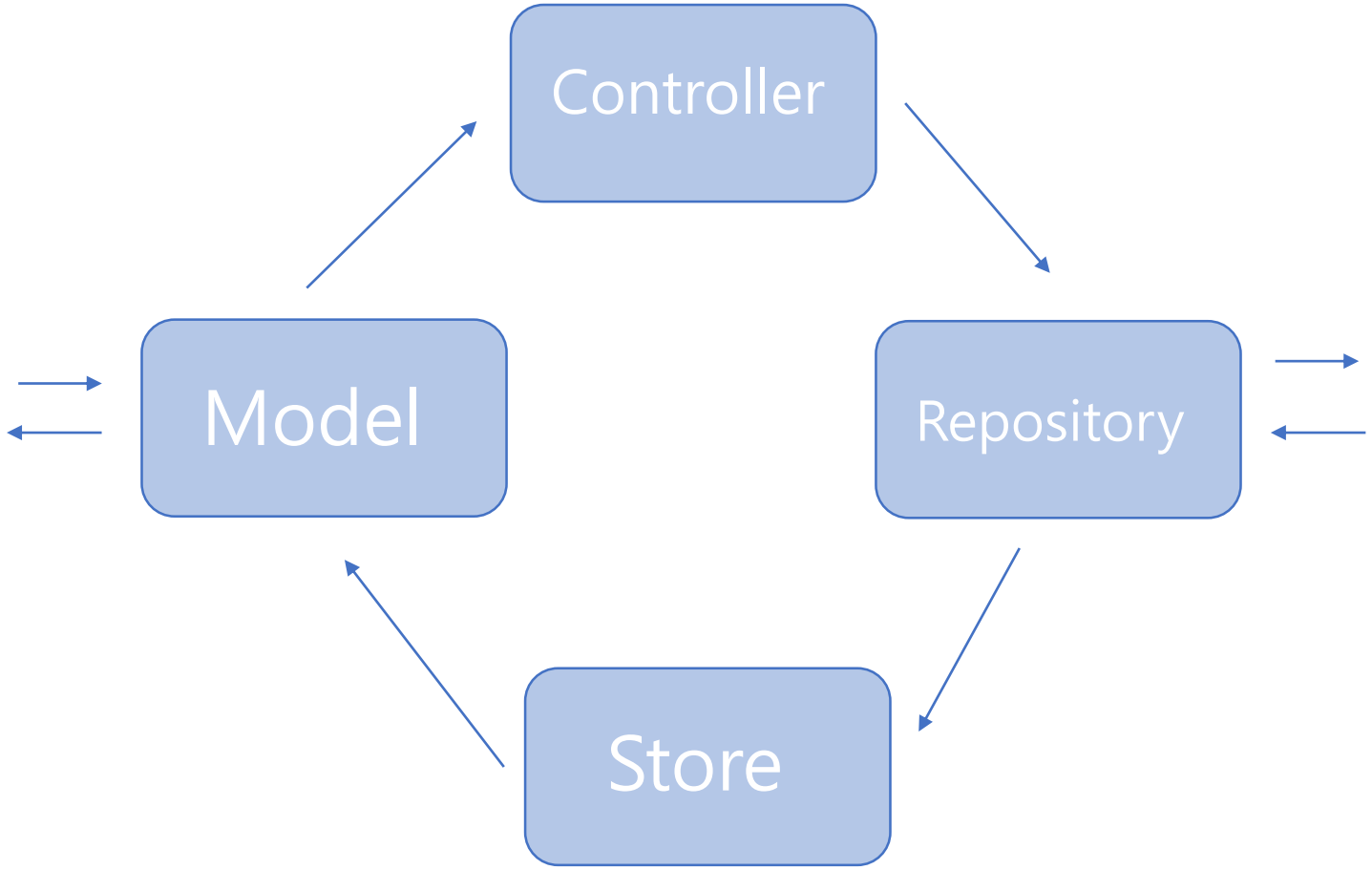
- 프로젝트 아키텍처
- 코드 컨벤션
- 적용 기술 소개

02 프로젝트 아키텍처

MVCS 아키텍처

MVC + S (store)

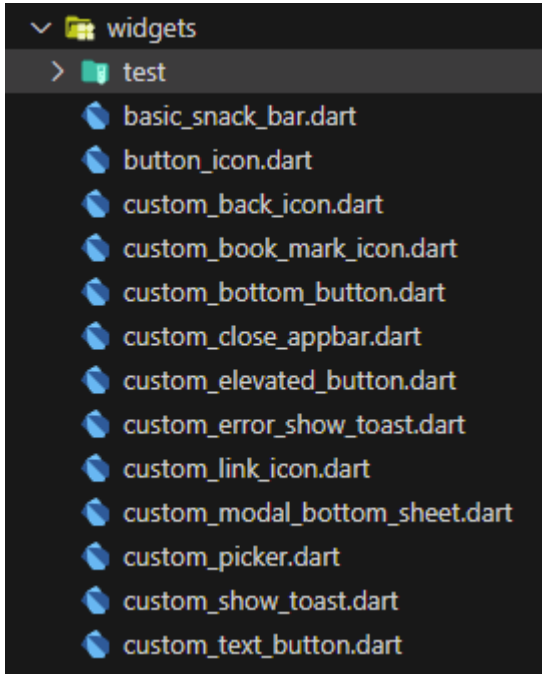
- 일반적인 MVC아키텍처에서 Store를 추가한 아키텍처
- 뷰에서 특정 이벤트를 발생시키면 컨트롤러에서 레파지토리를 통해 통신으로 데이터를 가져와 Riverpod 프로바이더(Store를 갱신)에 저장 한다.



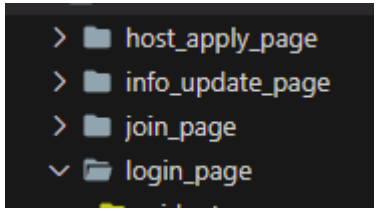
02 코드 컨벤션

코드 컨벤션

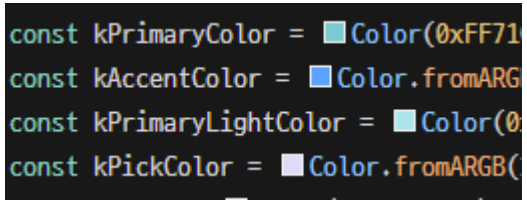
1. 공통적으로 쓰는 위젯들은 Wideget 폴더에 이름에 custom을 붙여서 만들어 놓고 사용



2. View 파일 끝에는 page를 붙이고 컴포넌트로 만들 수 있는 부분은 대부분 컴포넌트화



3. 색상에는 k를 붙여서 사용



02 적용 기술 소개

Flutter RiverPod

- Flutter 애플리케이션의 상태 관리와
  의존성 관리를 위한 라이브러리

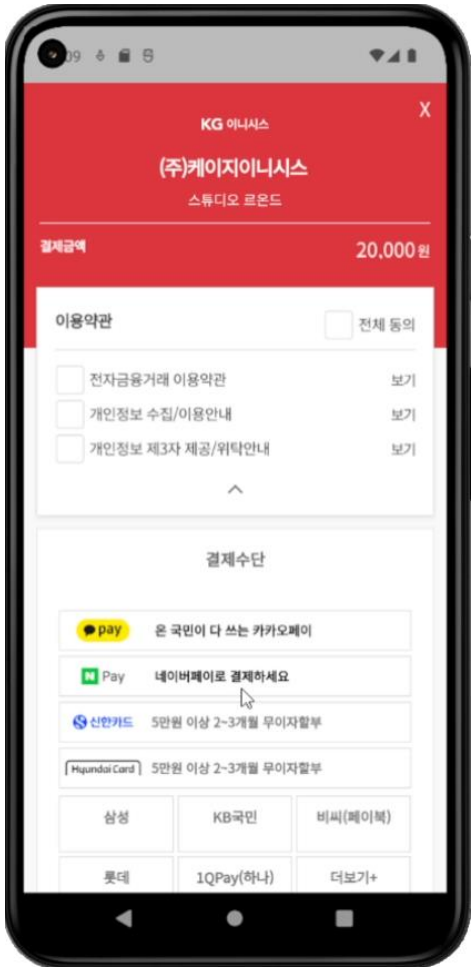
[장점]

- 변경된 상태만 감지하여 필요한 위젯만
  다시 렌더링하므로, 불필요한 렌더링을 피하고
  애플리케이션의 효율성을 향상시킨다.

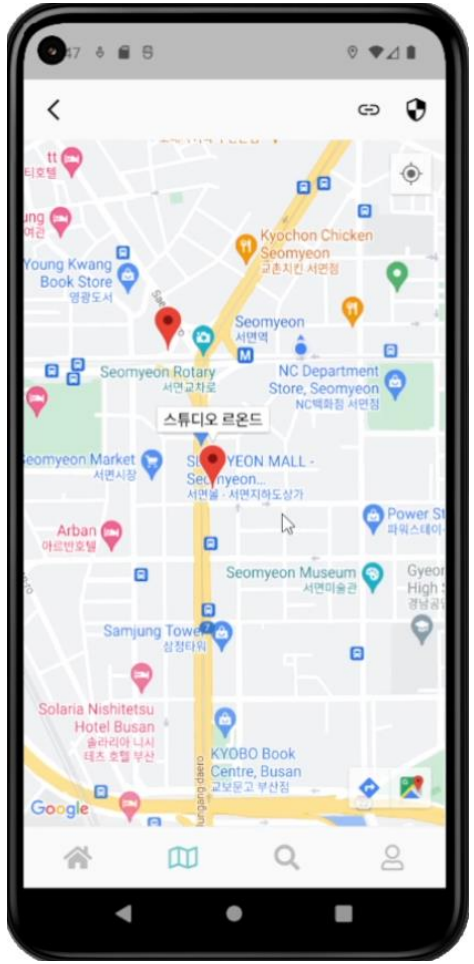
02 적용 기술 소개

API

Bootpay



Google Map

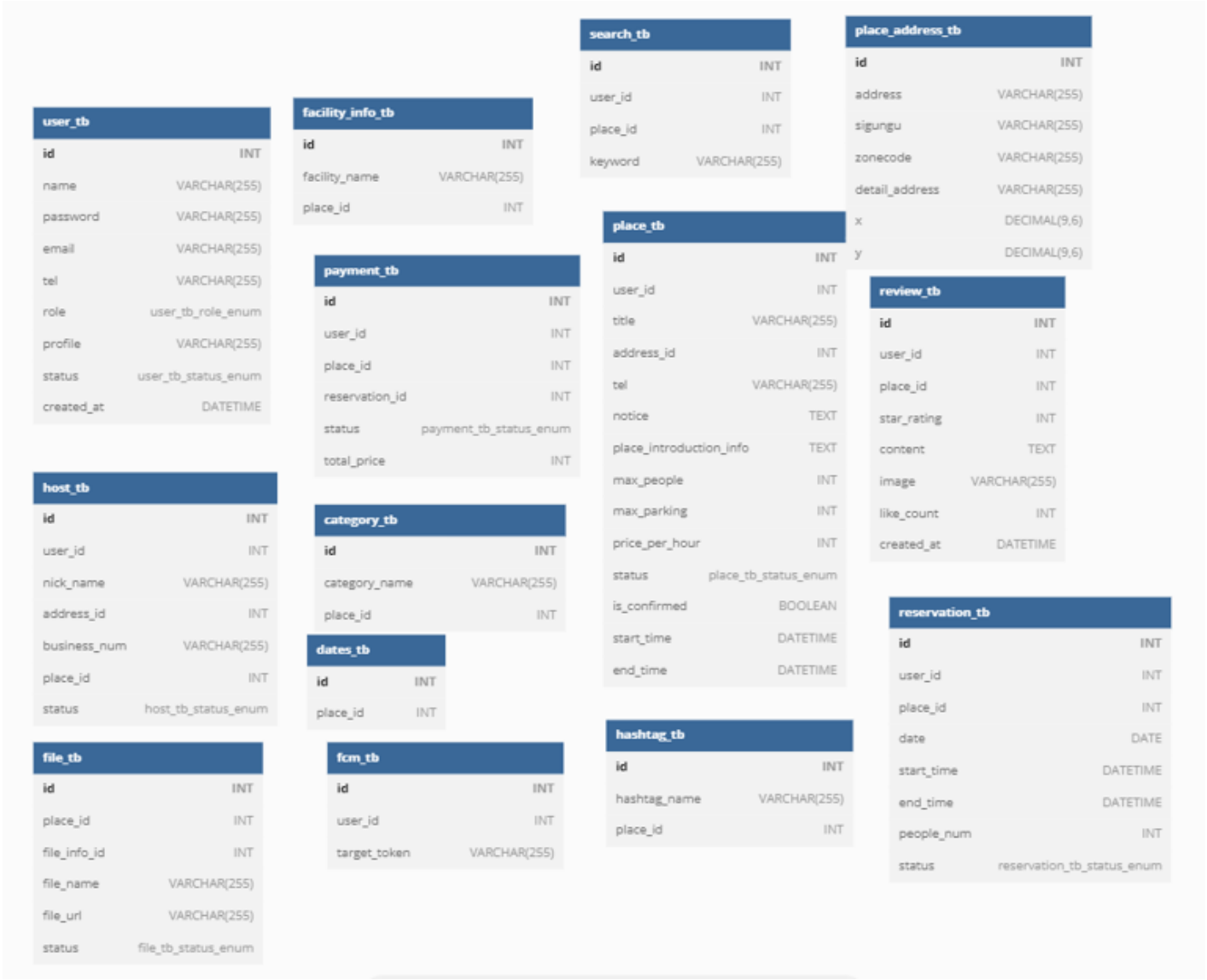


Firebase

03 백엔드

- 테이블 설계
- 프로젝트 아키텍처
- 코드 컨벤션
- 적용 기술 소개

03 테이블 설계



[주요 테이블]

유저 계정

공간 등록

예약 결제

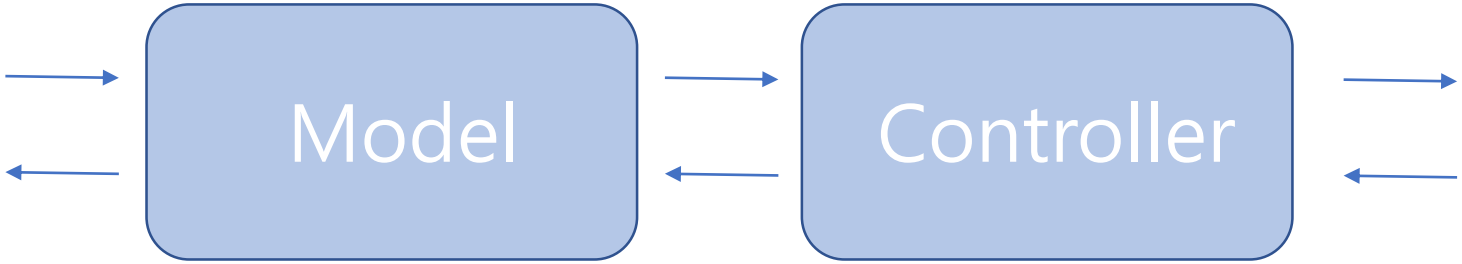
Host

이미지 저장

03 프로젝트 아키텍처

MC 아키텍처

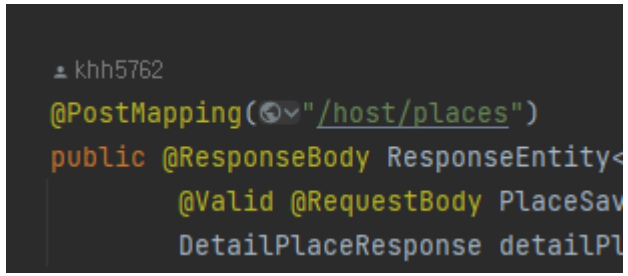
- RestAPI 서버를 만들어서 요청을 json으로 응답한다.



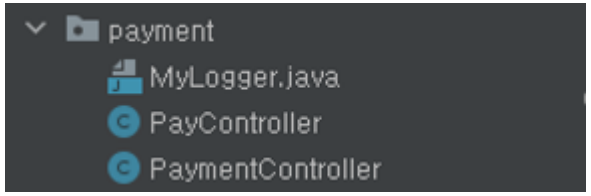
03 코드 컨벤션

코드 컨벤션

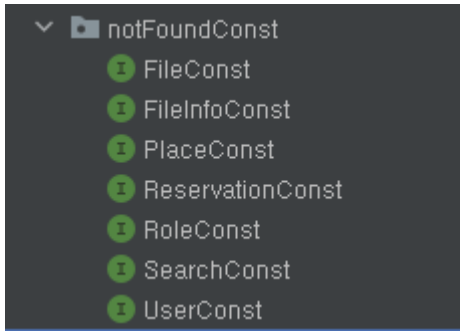
1. 권한이 필요한 경우 : /권한/내용



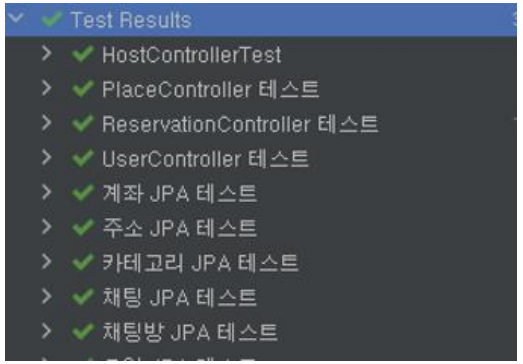
2. 클래스명 파스칼표기법 추가



3. 찾을 수 없는 exception을 사용 할 때 const에서 사용



4. Test완료 후 push



03 적용 기술 소개

Spring security + jwt

Spring Security

- 인증, 인가, 권한 부여, 세션 관리 및 보안 이벤트 처리와 같은 다양한 보안 기능을 제공합니다.
- JWT를 처리하는 기능을 기본으로 제공하므로, 개발자들은 JWT를 쉽게 사용할 수 있습니다.

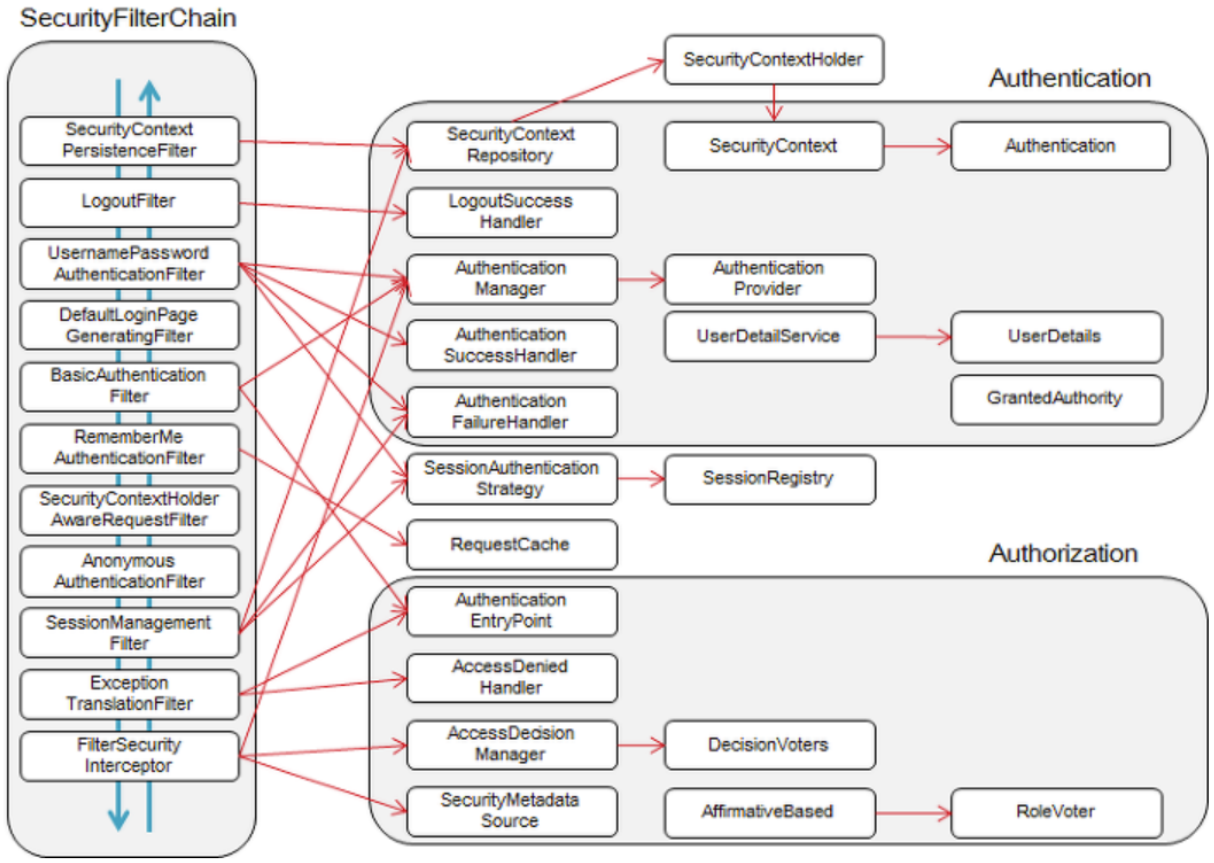
JWT

-서명되어 있으며, 서버에서 서명을 검증하고 토큰의 무결성을 보장합니다.

-따라서 JWT는 위조나 변조가 불가능한 안전한 방식으로 정보를 전달할 수 있습니다.

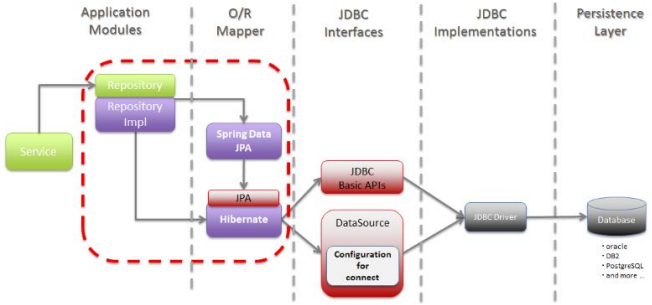
[장점]

- Spring Security에서는 spring security와 jwt를 사용하여 개발자가 쉽게 사용할 수 있어서 신뢰성있는 애플리케이션 개발이 가능하였습니다.



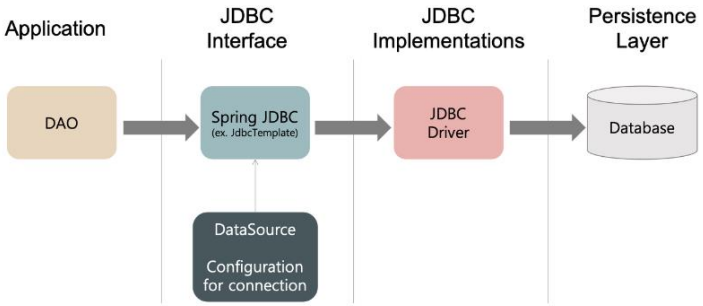
03 적용 기술 소개

JPA



-개발자가 데이터베이스와 상호작용할 때
필요한 복잡한 SQL 문법을 몰라도 되며,
더욱 객체 지향적이고 유지보수가 쉬운 코드를
작성할 수 있습니다.

jdbc templete



- JDBC API를 더 쉽게 사용할 수 있도록 해주며,
JDBC 작업에 필요한 반복적인 코드를 줄여줍니다.

03 적용 기술 소개

jpa와 jdbc를 함께 사용한 이유

- JPA는 JDBC를 직접 다루는 것보다 개발자가 코드를 적게 작성할 수 있으며,
  코드의 가독성과 유지보수성이 좋아집니다.

- 하지만 때때로 JPA만으로는 복잡한 쿼리사용이나 성능 튜닝이 어려운 경우가 있습니다.
  이런 경우에는 JPA와 함께 JDBC를 함께 사용하여 최적의 성능과 유연성을 얻을 수 있습니다.

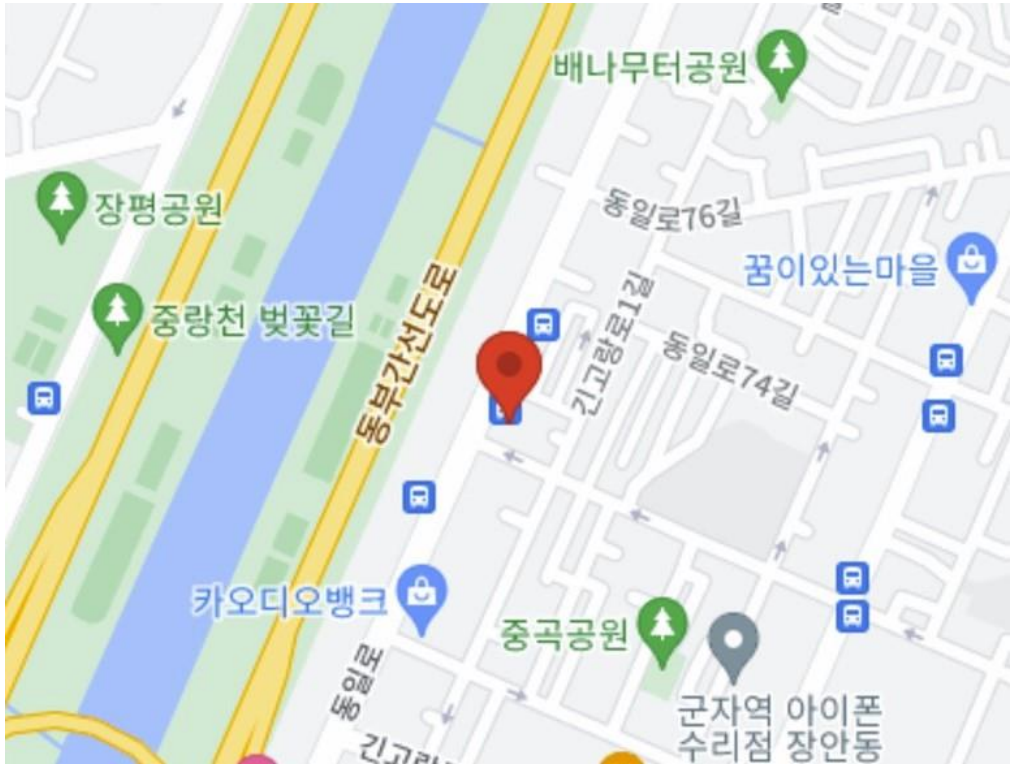
[장점]
- 따라서 저희는 기본적으로는 간단한 수행은 jpa를 사용하고
  복잡한 쿼리를 사용하는 검색과 필터에서는 jdbc templete를 사용하여
  데이터에 양과 순위만 조절해서 줄 수 있었습니다.

03 적용 기술 소개

Google Map API

-API키와 필요한 데이터를 파라미터로
요청하면 이미지 URL을 응답을 받습니다.

[사용 이유]
-장소를 등록하면 위치를 시각적으로 보여주고
싶어서 사용했습니다.



03 적용 기술 소개

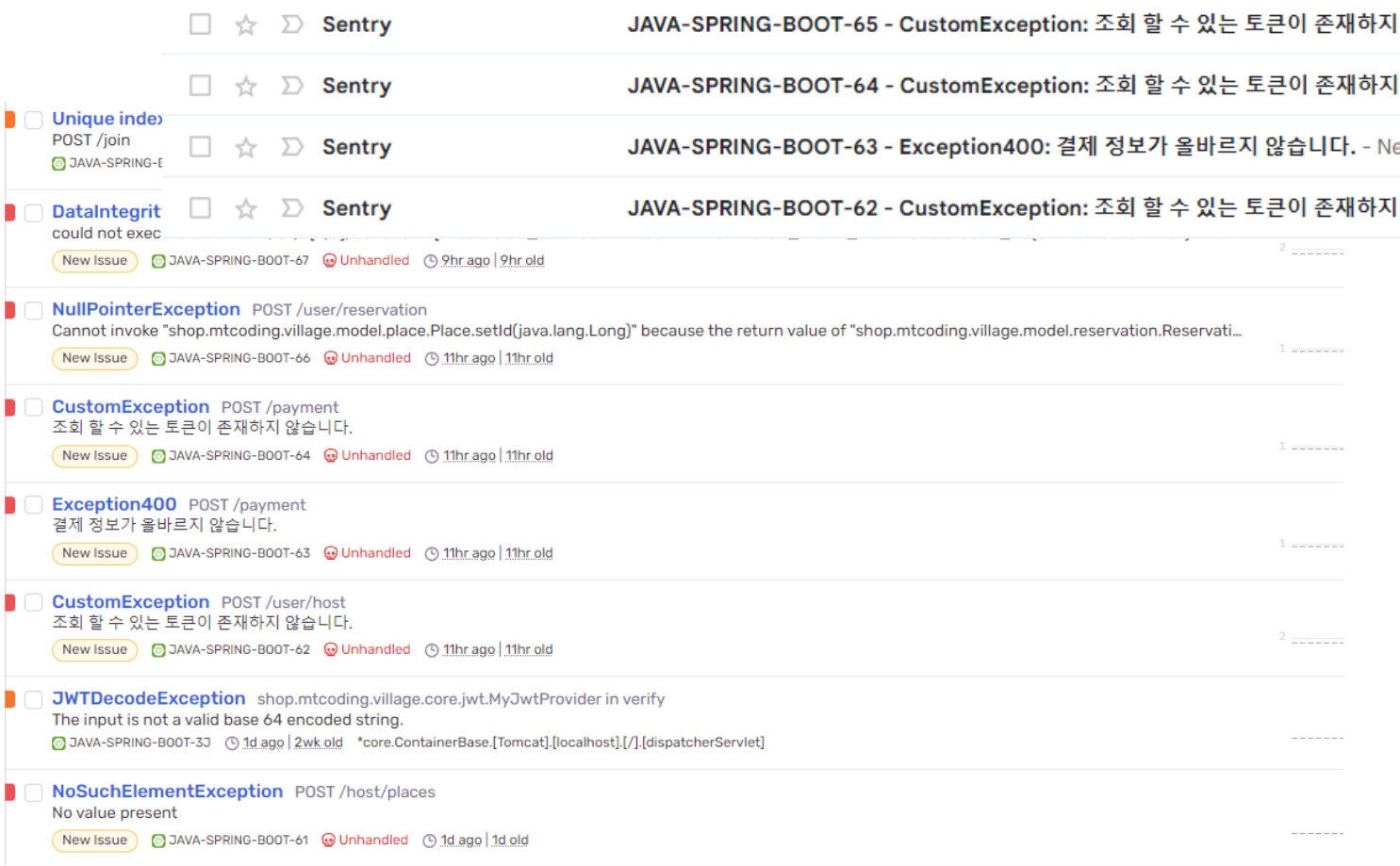
Sentry

[사용 이유]

-실시간으로 에러를 모니터링이 가능하여 대처를 빨리 할 수 있다.

-이메일이나 슬랙으로 알림이 와서 빠르게 대응 할 수 있다.

-Exception이나 오류. 로그들을 그룹화해서 빈도수 체크, 이벤트 그룹화 및 시각화해서 볼 수 있다.

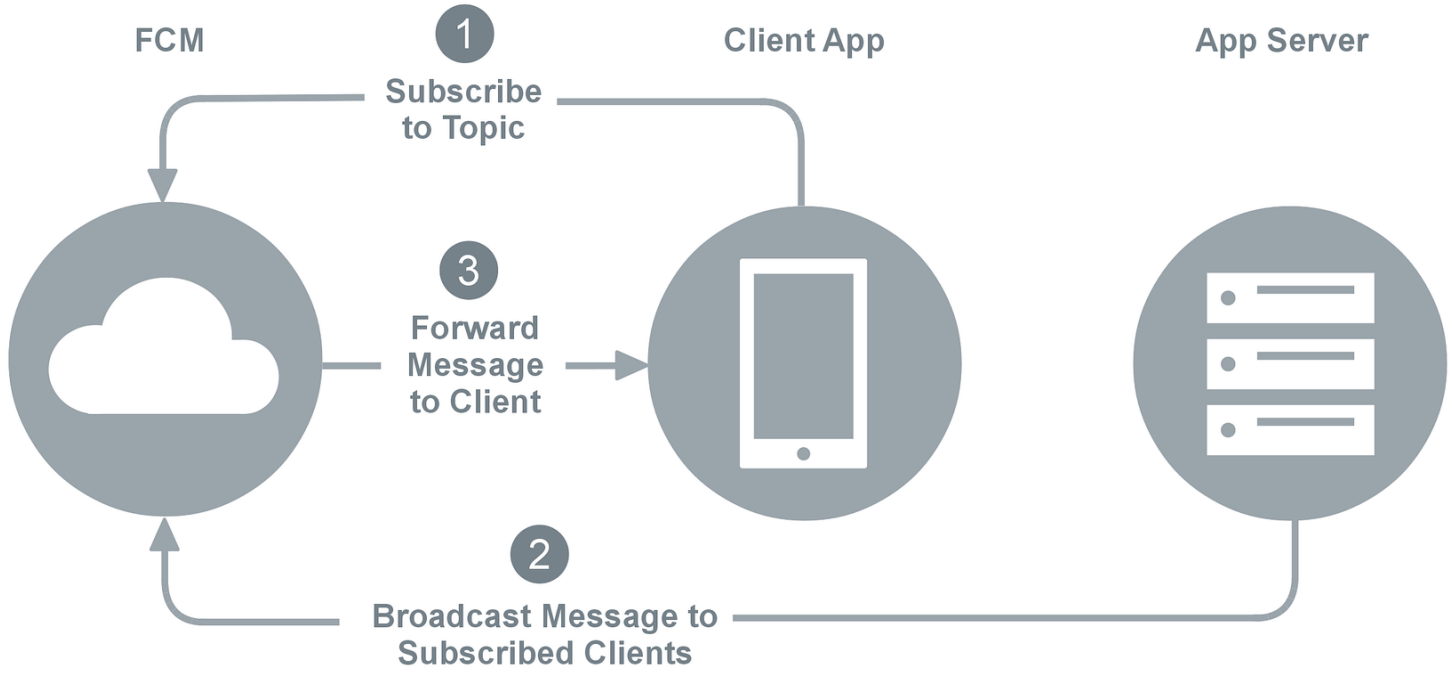


03 적용 기술 소개

파이어베이스의 Firebase Cloud Message

[사용 이유]

-실시간으로 수신이 이루어 지고
무료로 사용을 할 수 있는 푸시 알림으로
사용하기 편하기에 사용하였습니다.



03 적용 기술 소개

Rest doc

[사용 이유]

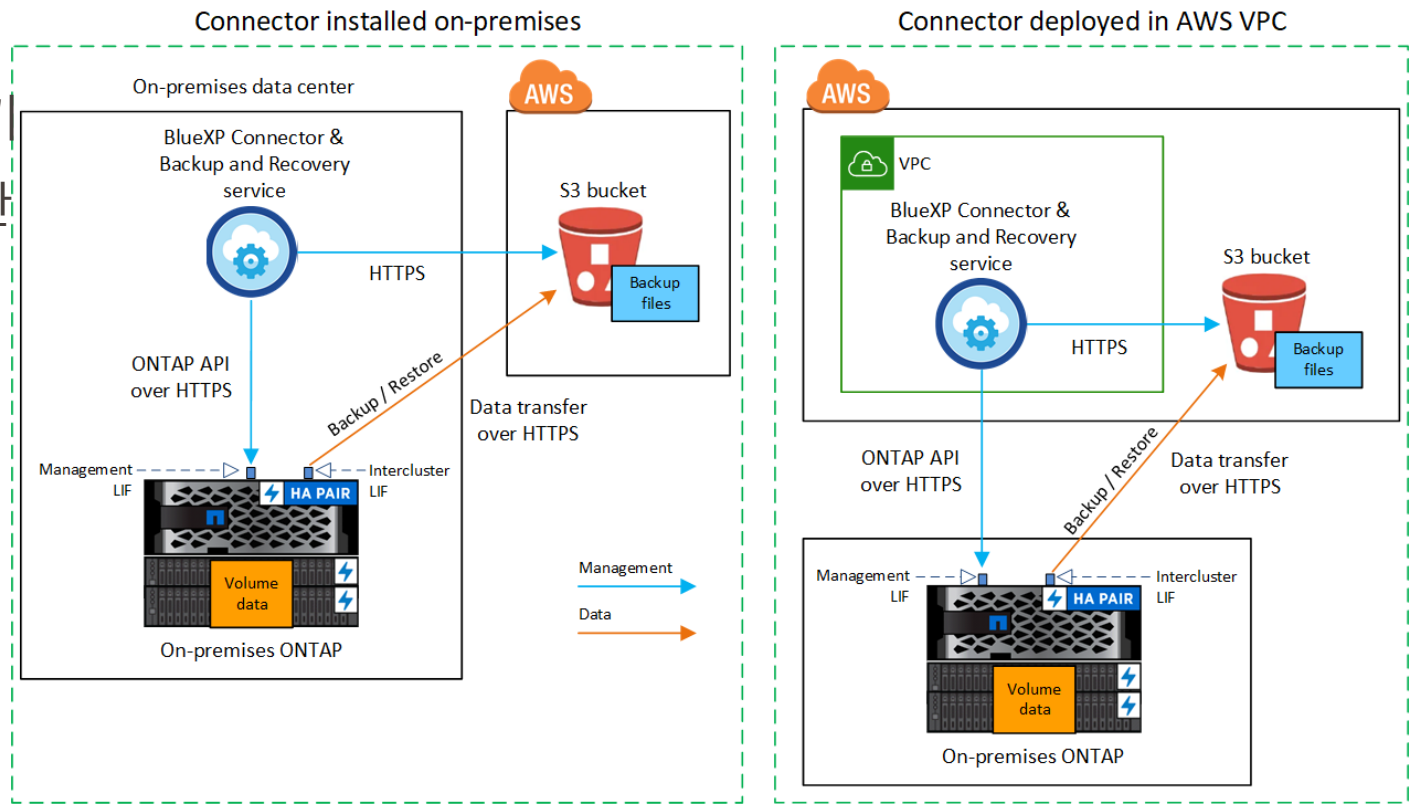
-Test코드를 작성하고 통과해야 문서가 나오기 때문에 따로 API를 문서를 만들 필요가 없기때문에 효율적이라 생각하여 사용하였습니다.

03 적용 기술 소개

AWS S3

[사용 이유]

- 안정성과 확장성이 있고 보안적으로 뛰어나기
때문에 사용을 해보고 싶었는데 문서를 보니 간
편히 사용을 할 수 있기에 사용하였습니다.



04 프로젝트 후기

-백엔드와 프론트엔드

-프로젝트 후기

04 프로젝트 후기 – 백엔드와 프론트엔드

[프론트엔드]

- 프로바이더의 생명주기가 짧아
  데이터가 null로 나와 해결한다고
  오래 걸렸다

- API문서를 만들고 작업을 했는데 계속 달라져서 작업 진행이 더뎌 졌다. 확실히 필요한 데이터를 정하고 작업해야 할 필요를 느꼈다.

[백엔드]

- Rebase전략에서 익숙하지 않아서
  충돌문제가 많이 발생

-EC2에서 bulid안됨
  bootjar로 만들고 EC2에 올렸는데 빌드가
  안되었다.

-Clean build를 사용
  테스트 오류까지 발견되어서 로그를 알려준다.

04 프로젝트 후기

박인우 (프)

프로젝트를 진행하면서 초기 계획과 많이 달라져 힘든 부분도 있었지만 조금 더 개발이라는 업무와 가까워진 것 같다
팀원들을 이끌면서 팀장이라는 역할에 대해 다시 한번 생각하는 계기가 되었다.

김정욱 (백)

배려심과 책임감이 강한 팀원들과 작업해서 저는 너무 행복하게 작업했습니다.
6개월동안 저도 그렇고 모두들 쉴틈없이 작업했는데
다들 원하는 곳에 취업하고 좋은 결과 얻었으면 좋겠습니다

김호현 (백)

프론트와 백엔드로 나누어 프로젝트를 처음 해봤기에 소통을 하는 부분이 부족하다는 것을 느꼈고 JPA를 처음 사용하면서 처음에는 복잡하고 힘들다고 생각을 했지만 잘 이해를 하면 오히려 편하게 사용할 수 있다고 생각이 들었고 무엇보다도 Restdoc을 사용하기 위해 test 코드를 작성을 하며 좋은 결과를 이루었다.

이인화 (프)

파이널 프로젝트를 하며 다양한 기능을 구현해 볼 수 있어서 재밌었습니다.
리버팟에 익숙하지 않아 데이터를 바인딩하는데 어려움을 겪어 아쉬웠지만 팀원들과 함께 프로젝트를 만들어 좋은 경험이었습니다.

[QnA]

감사합니다.