

Weekly Progress, Oct 11st, 2024

Housit- hows it?

:HouseCare Mobile Application

Team A : A+

Minsoo KIM, Seoyoung JO, Seoyoung JIN,
Jinwook CHOI, Sooyoung HWANG



Index

01 Role & Action Plan

02 What has been done & by whom this week?

02-01 Designing UX/UI and workflow of the application

- Specifying the features of app
- Designing/fixing UX/UI
- Designing prototype of app with Figma

02-02 Frontend & Backend preparation & set-up

- Designing ERD(Entity Relationship Diagram)
- API documentation

02-03 OCR Feature Implementation

- Technical Background

03 What will be done & by whom next week?

01 Role & Action Plan

Name	Role
김민수	Planning, Document, UI/UX
조서영	OCR Feature Implementation
진서영	UI/UX Design
최진욱	Frontend
황수영	Backend

[illegible]

02 What has been done & by whom this week?

02-01 Designing UX/UI and workflow of the application

- Specifying the features of app @김민수
- Designing/fixing UX/UI @김민수 @진서영
- Designing prototype of app with Figma @진서영

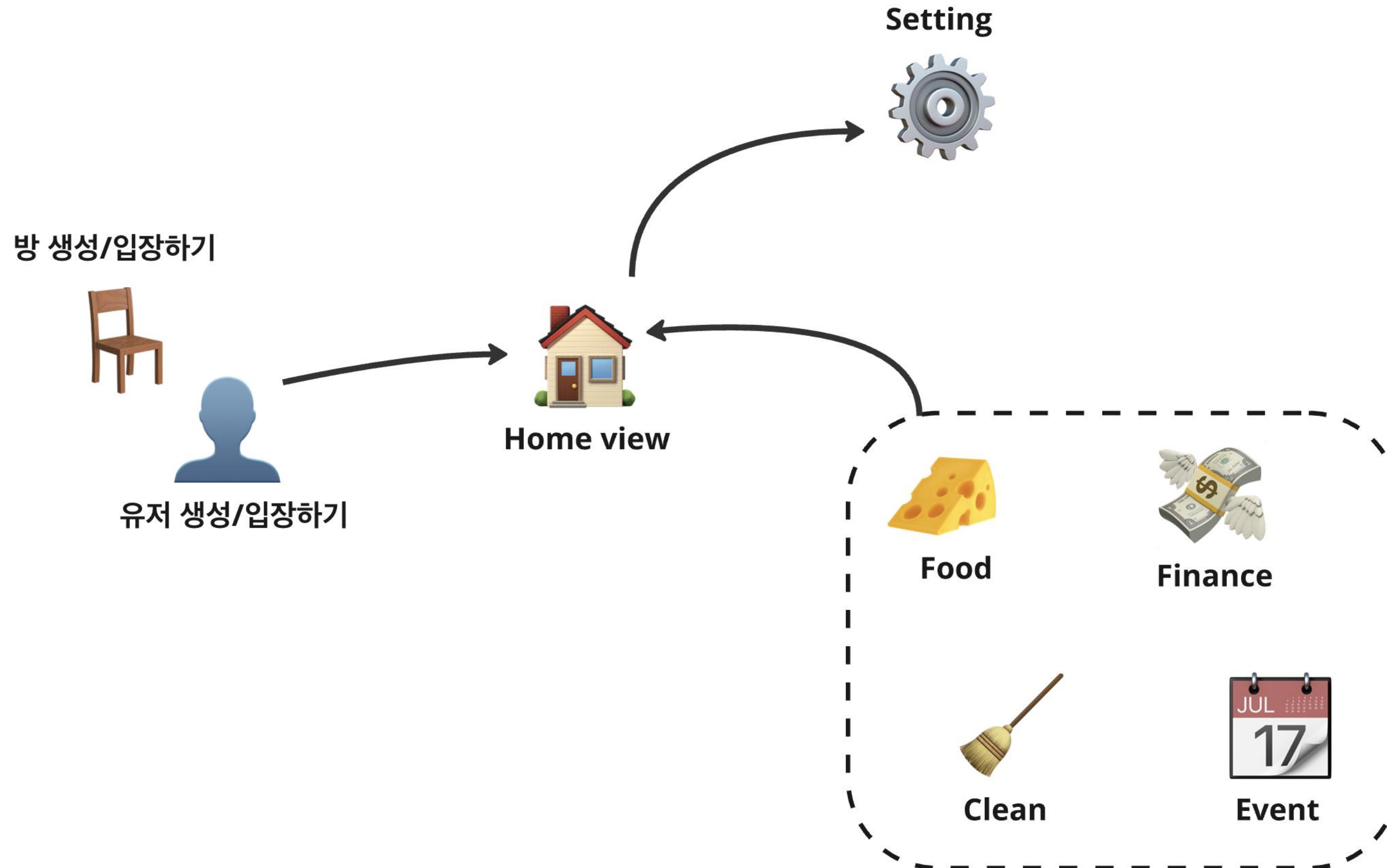
02-02 Frontend & Backend preparation & set-up

- Designing ERD(Entity Relationship Diagram) @황수영
- API documentation @황수영 @최진욱 @김민수
- Basic setup(Frontend) @최진욱

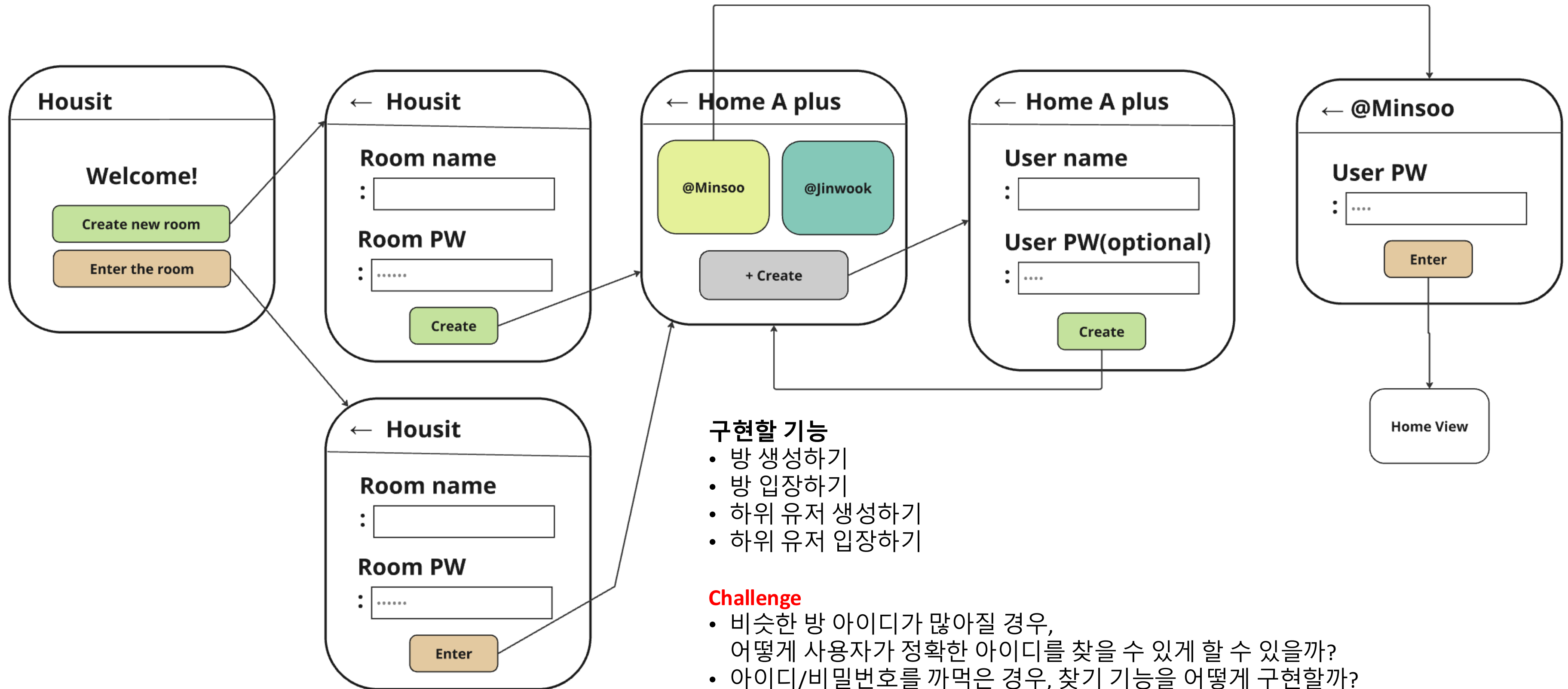
02-03 OCR Feature Implementation @조서영

- Architecture configuration of how to implement OCR functionality
- Buying a IP camera

02-01 Main features of our app - @김민수



Feature Details : Room/User



Feature Details : Home

구현할 기능

- Food에서 가져올 정보
 - 01 유통기한이 임박한 식품
 - 02 좋아하는 음식이 설정한 갯수 이하로 떨어질 경우
- Finance에서 가져올 정보
 - 날짜가 임박한 예상 지출 (예: Netflix 결제, 10월 12일(토))
- Clean에서 가져올 정보
 - 날짜가 임박한 청소 담당구역 (예: 빨래, @민수, 10월 12일(토))
- Event에서 가져올 정보
 - 날짜가 임박한 공동 일정

Challenge to discuss

- Food/Finance/Clean/Event에서 임박한 정보를 가져와 띄울 때, '당일' 뿐만 아니라, 'N일 전'을 사용자가 설정할 수 있게 하면 좋겠음!



Feature Details : Food

Sorted by Expiration Date Date

Food

▼ Expiration Date

Frozen Foods(냉동)

Chicken Breast : 20

Purchased Date : 2024/9/23(Mon)

@all

Pizza : 1

Purchased Date : 2024/9/23(Mon)

@Mike

Blueberry :

50%

Purchased Date : 2024/9/18(Wed)

@all

Refrigerated Foods(냉장)

Yogurt : 6

Expiration Date : 2024/10/7(Sun)

Purchased Date : 2024/10/3(Thu)

@Mike

Seoul milk : 2

Expiration Date : 2024/10/10(Thu)

Purchased Date : 2024/10/3(Thu)

@all

Cheese : 4

Expiration Date : 2024/10/24(Thu)

Purchased Date : 2024/10/3(Thu)

@jisoo

Room Temperature Foods(상온)

Rice :

50%

Purchased Date : 2024/9/18(Wed)

@all

+ Add item

Modifying information of a food

Food

▼ Purchased Date

Refrigerated Foods(냉동)

Chicken Breast : 20

Purchased Date : 2024/9/23(Mon)

@all

minimum quantity : 2

Name : Seoul milk

Quantity :

-

 2

+

▼ Count

Where? : ▼ Refrigerated Foods

Whose? :

@all

@jisoo

@Mike

Purchased Date : 2024/10/3(Thu)

Expiration Date : 2024/10/10(Thu)

Save

Cancel

Delete

Room Temperature Foods(상온)

Rice :

50%

Purchased Date : 2024/9/18(Wed)

@all

Spam : 5

Purchased Date : 2024/9/23(Mon)

@all

Display 'Near expiration date' food with card view

Food

▼ Purchased Date

Near Expiration Date

Yogurt : 6

4 days left

@Mike

Seoul milk : 2

7 days left

@all

Frozen Foods(냉동)

Chicken Breast : 20

Purchased Date : 2024/9/23(Mon)

@all

Pizza : 1

Purchased Date : 2024/9/23(Mon)

@Mike

Blueberry :

50%

Purchased Date : 2024/9/18(Wed)

@all

Refrigerated Foods(냉장)

Seoul milk : 2

Expiration Date : 2024/10/10(Thu)

Purchased Date : 2024/10/3(Thu)

@all

Cheese : 4

Expiration Date : 2024/10/24(Thu)

Purchased Date : 2024/10/3(Thu)

@jisoo

Yogurt : 6

Expiration Date : 2024/10/3(Thu)

Purchased Date : 2024/10/3(Thu)

@Mike

Room Temperature Foods(상온)

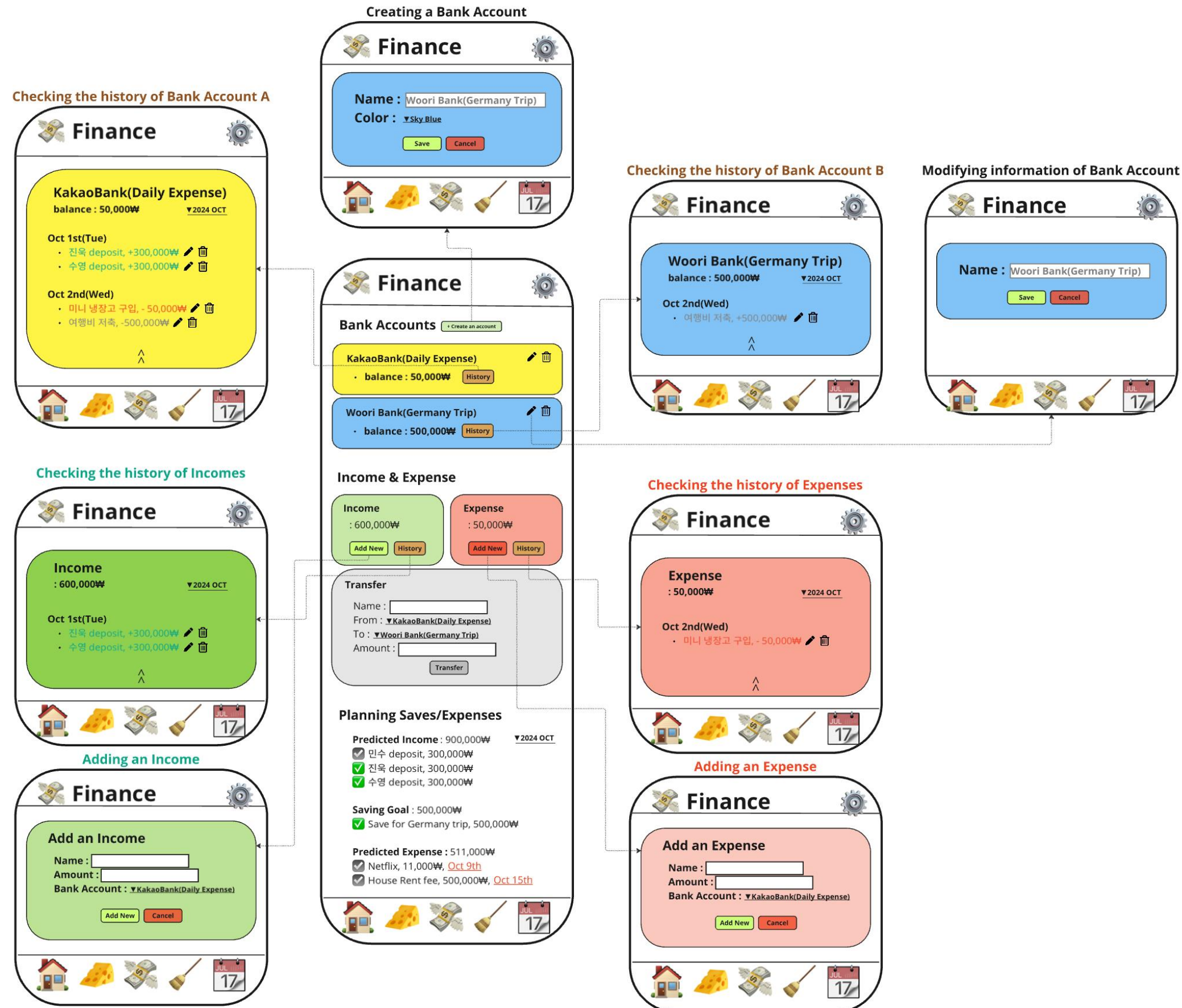
구현할 기능

- View
 - Grouping
 - 기본값 : '냉장/냉동/상온'으로 대분류
 - 음식 주인별로 묶음
 - 좋아하는 음식
 - Filtering
 - 좋아하는 음식만
 - Sorting
 - 유통기한(가까운 순)
 - 구입일자(오래된 순)
 - 유통기한 임박한 식품 상단 표시
 - 식료품 추가/수정/삭제
 - 양 표기 : 갯수 카운트 or 퍼센트
 - 공용식품 / 개인식품 태그
 - 공동 좋아요 기능
 - 설정한 최소필요 개수 이하로 떨어지면 알림

Challenge

- 아직 없음

Feature Details : Finance



구현할 기능

- `통장(계좌)` 잔금 파악
 - 통장 생성/수정/삭제
 - 월별 수입/지출/이체 내역 확인
- `수입/지출/이체` 파악
 - Transaction(수입, 지출, 이체) 생성/수정/삭제
 - 월별 수입, 지출 총액 & 내역 확인
- `예상 수입/지출/저축` 계획
 - 체크박스, 이름, 금액, 예상날짜 설정

Challenge

- 통장별로 색상 지정 기능 필요 -> 팔레트 Preset or #FFFFFF 고민중
- `예상 수입/지출/저축` 계획 항목이 리스트뷰로 display.
크기가 작아서 아이템마다 수정버튼을 달기보단,
전체편집 후 저장하는 UX가 더 사용성 좋을 듯함!
- `수입/지출/이체` 파악 버튼 통합 및 간소화 필요
- 모달(팝업)이 많아서 조잡 -> 다른 UI/UX 구상 필요(Bottom Sheet)

Feature Details : Clean

Creating a cleaning area(task)

Clean

Create a Cleaning Area

Name : Bathroom

종류 : ▾주방 화장실 세탁
Mon Tue Wed Thu Fri Sat Sun

Which day? :

How often? : ▾everyday
every week
bi-weekly
every month
every year(매도 될듯)

Who? : @all Minsoo Jinwook Sooyoung

Create Cancel

• Minsoo : Living Room, Toilet
• Jinwook : Kitchen, Laundry

Random

Clean View

Clean

Cleaning Area + Create

Living Room
Next day : Oct 19th(Sat) 2024

Kitchen
Next day : Oct 19th(Sat) 2024

Laundry
Next day : Today

Toilet
Next day : Nov 1st(Fri) 2024

Updated : 10/8(Tue)

• Minsoo : Living Room, Toilet
• Jinwook : Kitchen, Laundry

Random

Randomizing & Assigning cleaning areas(tasks)

Clean

Random Roulette

Members : @all Minsoo Jinwook Mike

Cleaning Area
☒ Living room
☒ Kitchen
☒ Laundry
☒ Toilet

Spin! Cancel

October 2024

• Minsoo : Living Room, Toilet
• Jinwook : Kitchen, Laundry

Random

Checking the randomized result

Clean

Random Roulette

Result

• Minsoo : Living Room, Toilet
• Jinwook : Kitchen, Laundry

Spin Again? Save

Cancel

October 2024

• Minsoo : Living Room, Toilet
• Jinwook : Kitchen, Laundry

Random

구현할 기능

- 청소구역 생성/수정/삭제
 - 청소 주기 & 요일 설정
 - 담당자 지정

- 청소구역 랜덤 룰렛 돌리기
 - 멤버 & 청소구역 선택 후 Spin

Challenge

- 카드 뷰 청소구역 픽토그램 어떻게 설정?
 - > 청소구역 생성/수정시 '종류' 선택 -> 이미지 설정
- 청소구역 순서 바꾸기
 - > 꺾 눌러서 이동한다면, 저장 어떻게?
 - > 순서 바꾸기 모달 추가?

Feature Details : Event




구현할 기능



- 이벤트 생성/수정/삭제
- 미래>현재>과거 순으로 이벤트 보여주기



Challenge to discuss


- 이벤트의 개수가 많을 경우, 한 번에 모든 이벤트 불러올 필요 없음
-> 아래로 스크롤하면 1분기(3달) 단위로 더 로딩하도록!
- Idea) 과거>현재>미래 순으로 보여주는데,
앱을 열면 현재 시점의 위치로 자동으로 띄울 수 있을지?


Feature Details : Setting

 **Setting**

 **Food**
 Notification for Expiration date
: 3 days before


 **Finance**
 Notification for Predicted Transaction
: 1 days before






 **Clean**
 Notification for Cleaning Task
: 5 days before


 **Event**
 Notification for Event
: 3 days before


Save


Cancel

 **Members**

 **Setting**

 **Members**

@Minsoo 

@Jinwook


Expel

Master






@Sooyoung

Expel

Master

 **Room**
Current Password :
Change Password :

Change

구현할 기능

- Food/Finance/Clean/Event 알림 'N일 전' 설정
- 무단침입자 밴(추방) 기능 & 방장 위임 기능
- 방 비밀번호 변경

Challenge to discuss

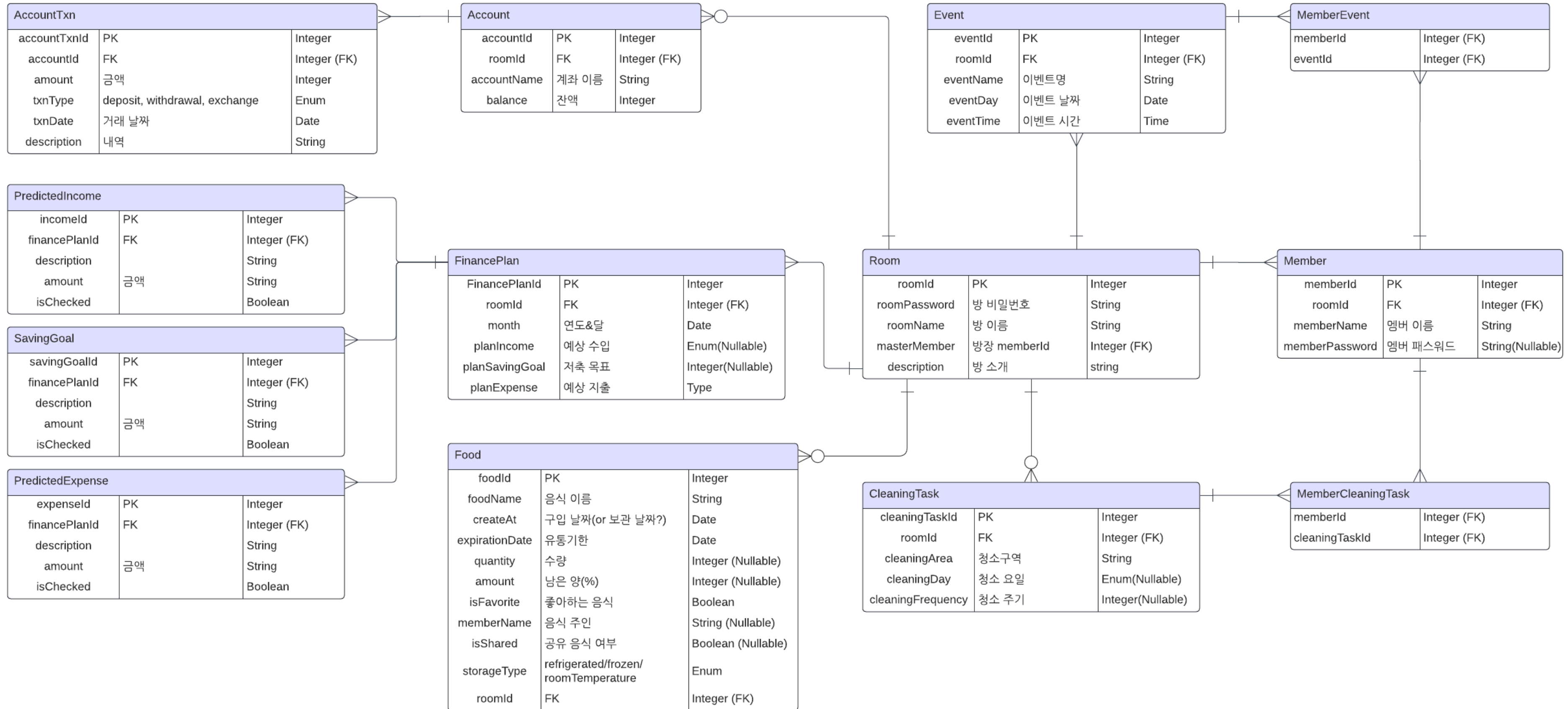
- 무단침입자 추방시, 방 비밀번호 필수로 변경
-> 중간에 앱 종료하면 어떻게 핸들링?

Prototype of app designed with Figma - @진서영
















ERD 설계 - 황수영

고려중인 사항
멤버 테이블에 유저 세팅 필드 추가



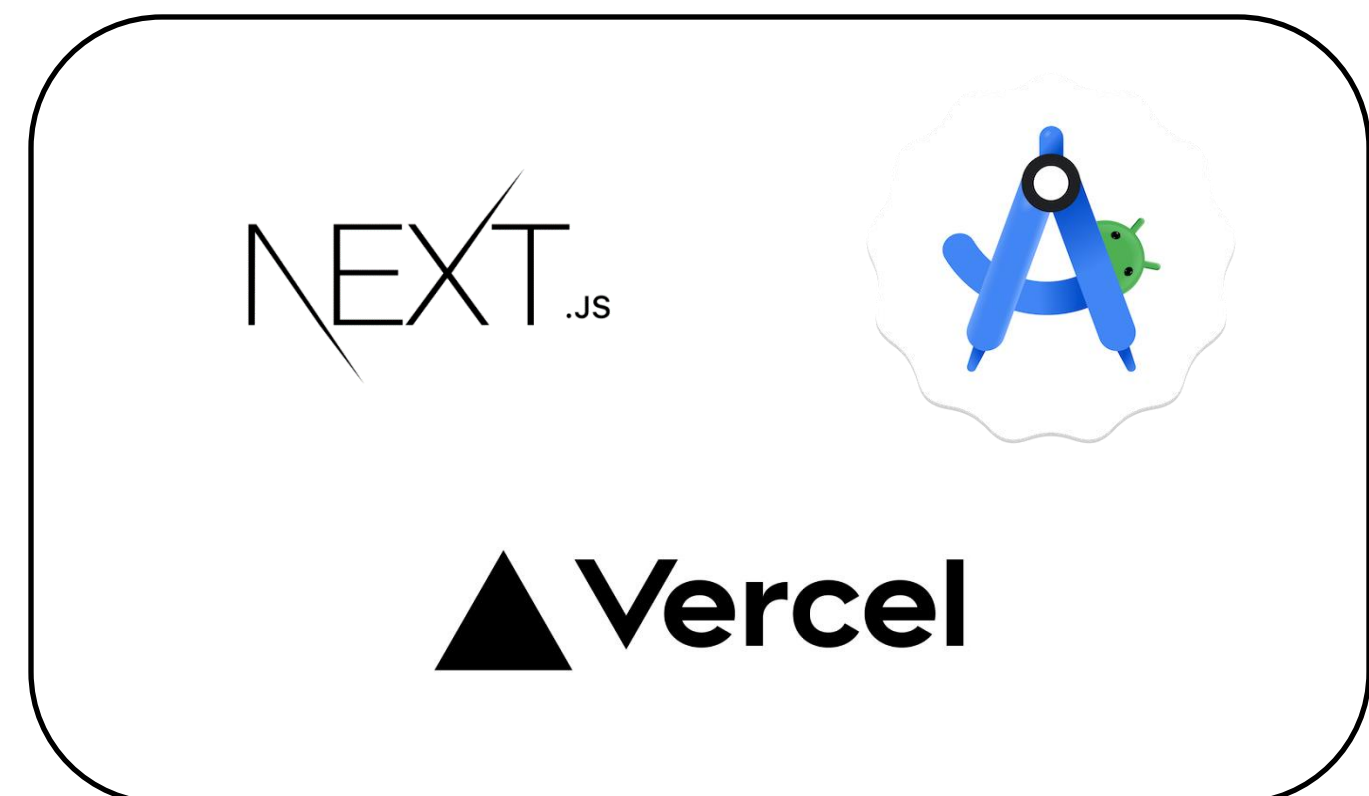
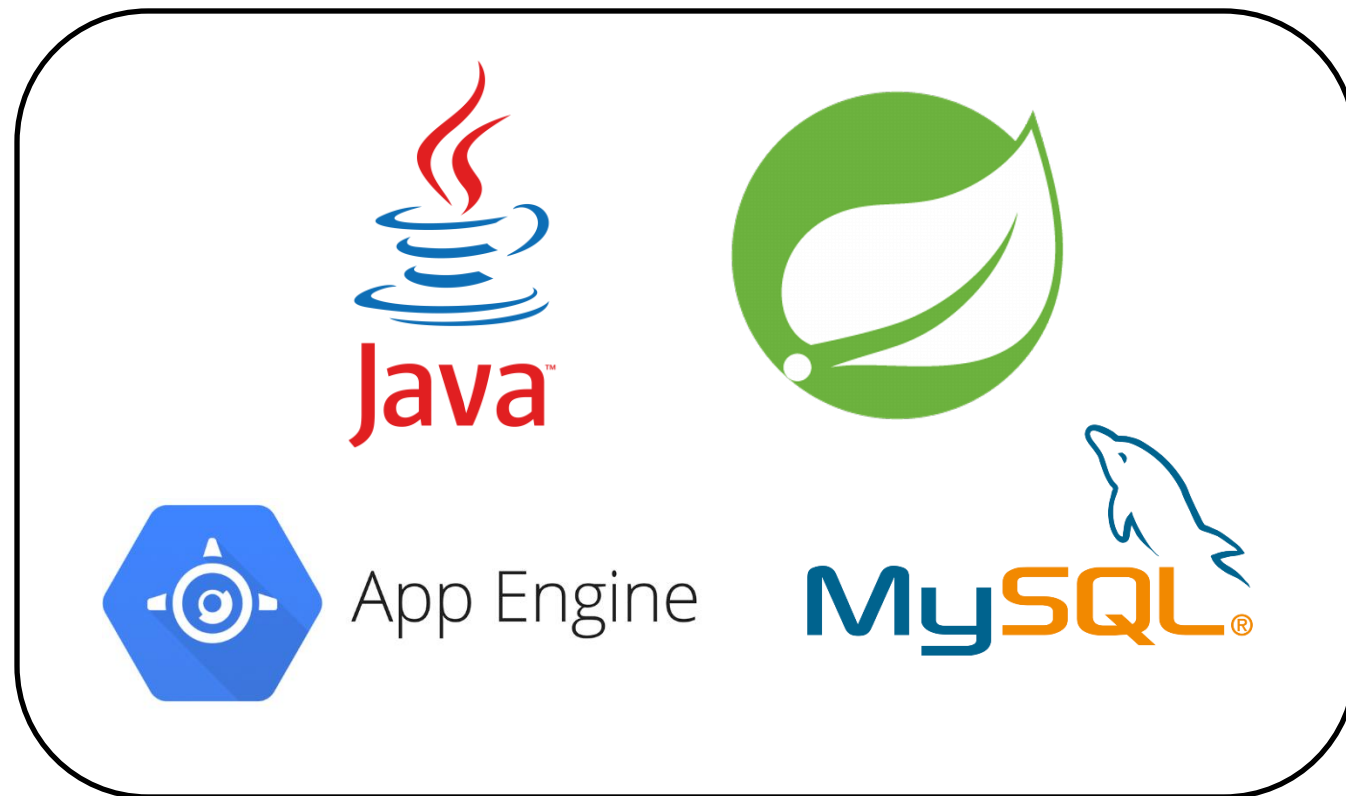
API 명세서 - @최진욱 @황수영 @김민수

 방 생성	POST	/room	
 방 입장	GET	/room/{roomId}	
 멤버 생성	POST	/room/{roomId}/member	
 음식 불러오기	GET	/room/{roomId}/foods	
 음식 등록하기	POST	/room/{roomId}/foods	
 음식 수정하기	PUT	/room/{roomId}/foods/{foodId}	
 이벤트 불러오기	GET	/room/{roomId}/events	
 이벤트 등록하기	POST	/room/{roomId}/events	
 이벤트 수정하기	PUT	/room/{roomId}/events/{eventId}	
 청소 불러오기	GET	/room/{roomId}/cleans	
 청소 등록하기	POST	/room/{roomId}/cleans	
 청소 수정하기	PUT	/room/{roomId}/cleans	
 finance 불러오기	GET	/room/{roomId}/finance? year=xxxx&month=xx	

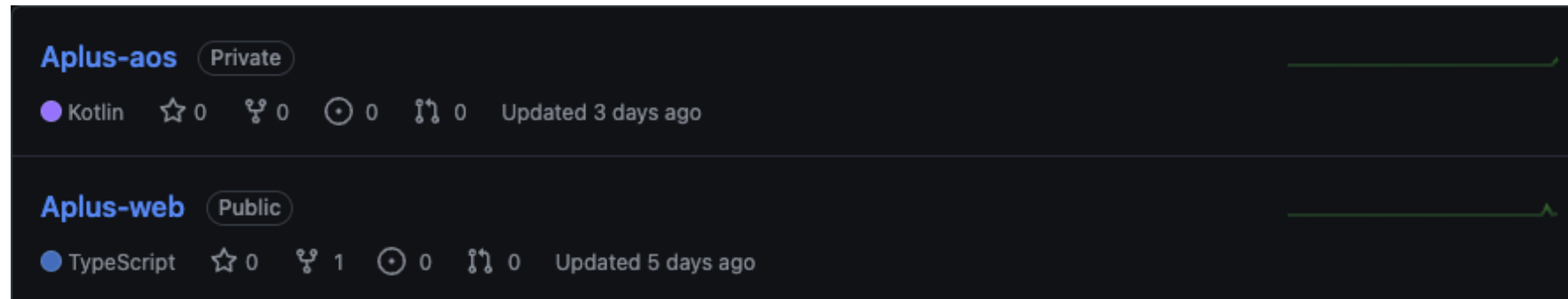
- Notion을 사용해 정리
- 백엔드 개발 진행하면서 Swagger로 자동화 고려 중
-> Postman, Swagger 둘 중 고민 중

02-03 Frontend & Backend set-up

- @황수영 @최진욱



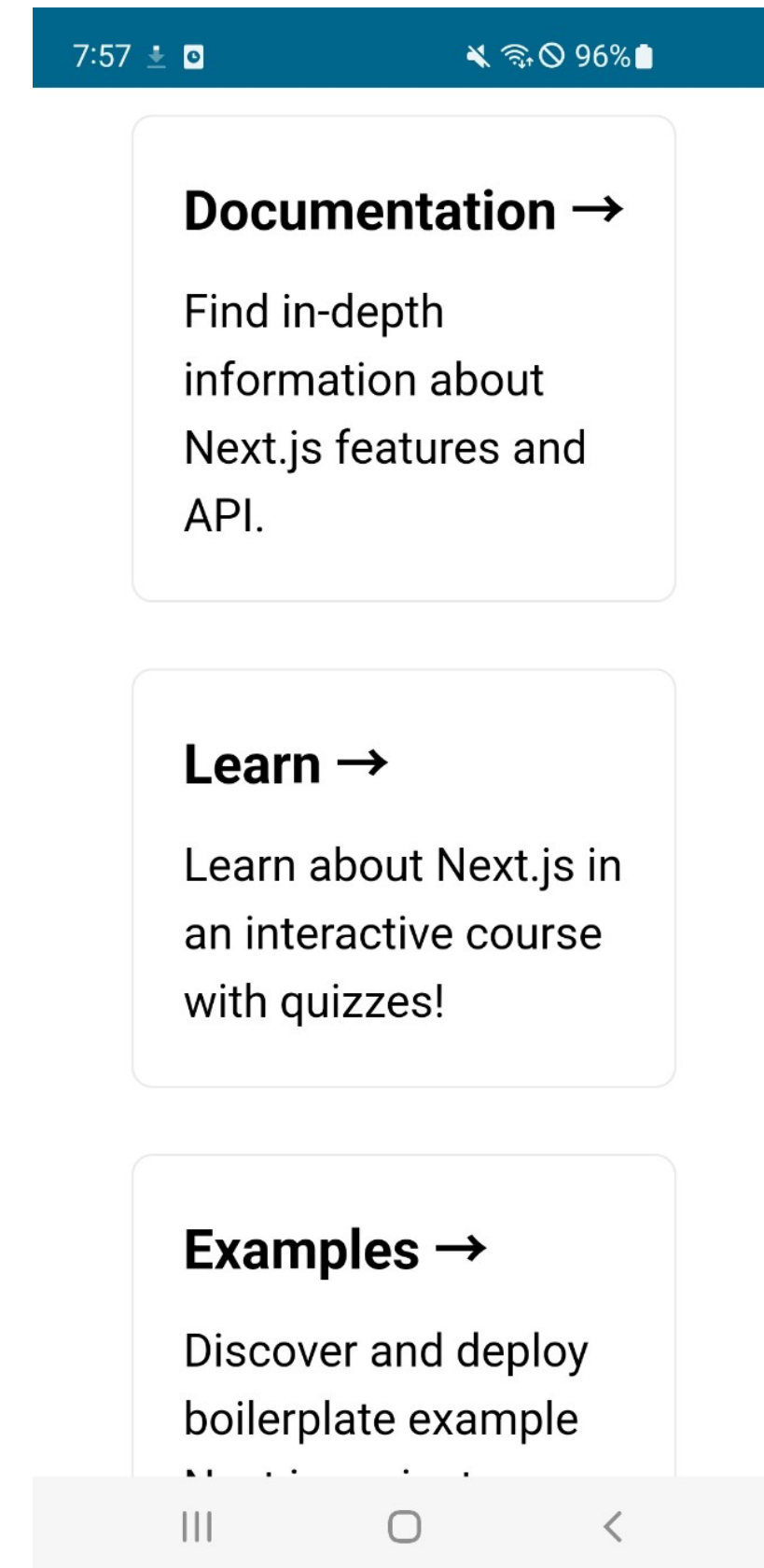
Basic setup @최진욱



웹뷰를 위한 Aplus-web, 앱 개발을 위한 Aplus-aos 두 레포로 관리

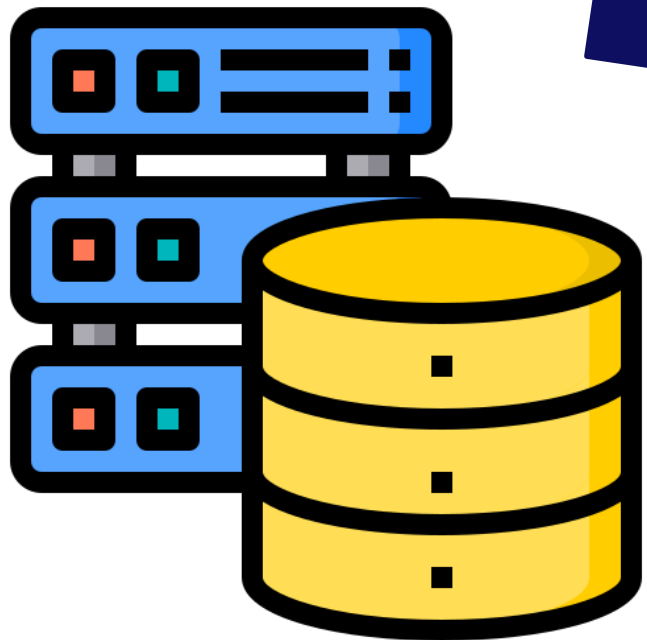
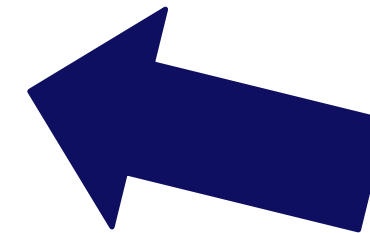
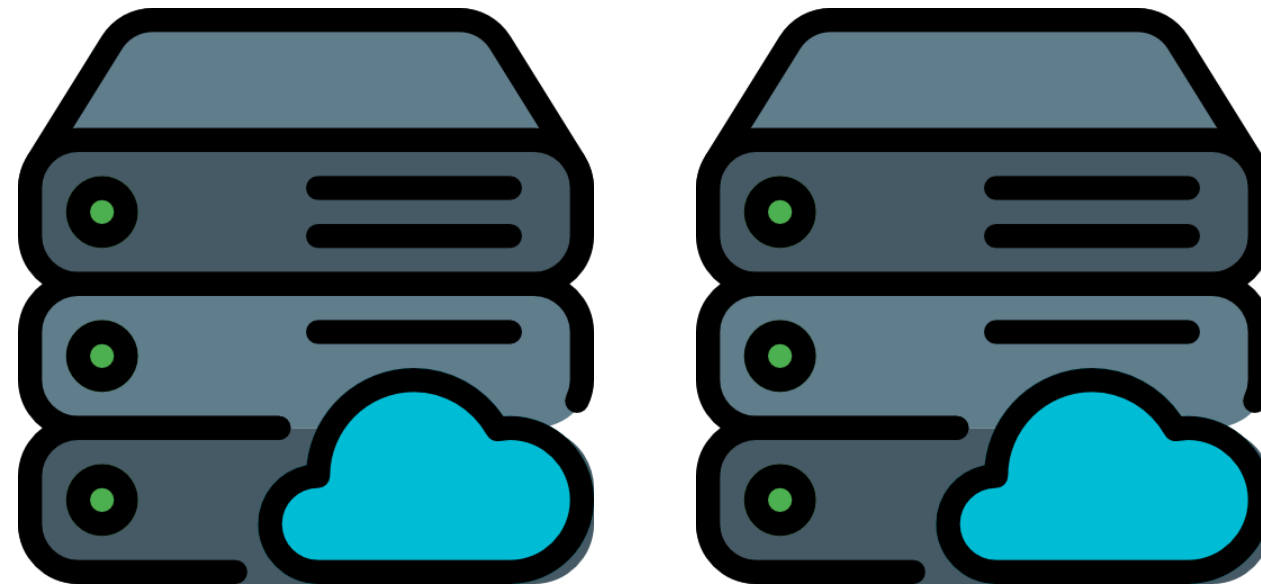
Aplus-web: 필수 패키지 설치 및 Vercel을 통한 배포 완료

Aplus-aos: 배포된 url을 통해 Webview 세팅 완료, 테스트용 안드로이드 폰으로 간단한 동작 테스트 중



02-03 OCR 기능 구현 계획 - @조서영

```
{ "foodname" : "서울우유",  
  "expirationDate" : "2024.12.31" }
```



(1) Spring 서버

- OpenCv 사용해서 비디오 프레임 캡처
- tess4j 사용해서 캡처 프레임에서 제품명과 유통기한 인식

(2) Flask 서버

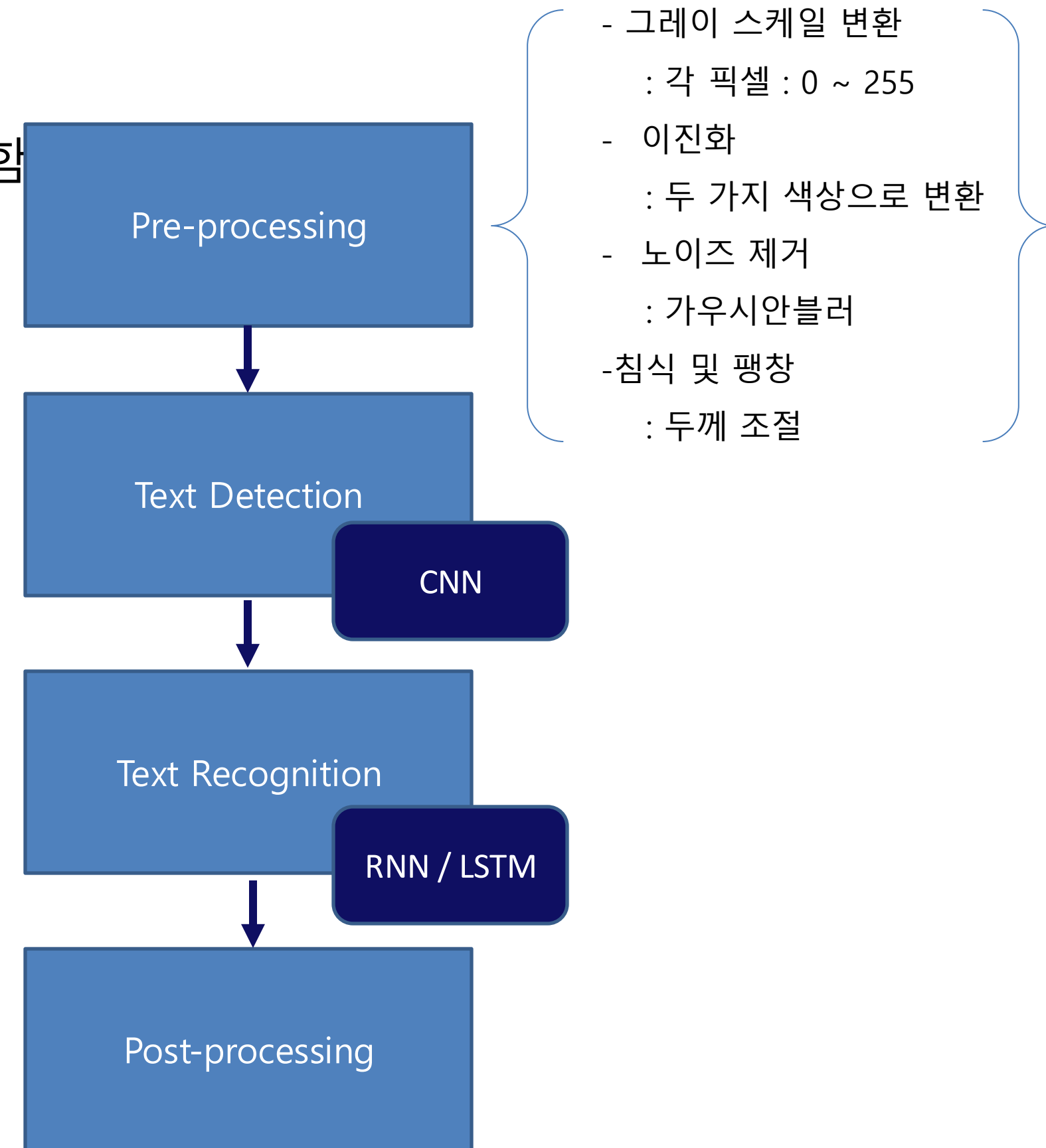
- OpenCv 사용해서 비디오 프레임 캡처
- pytesseract 사용해서 캡처 프레임에서 제품명과 유통기한 인식
- POST method를 사용해 spring 서버로 인식된 제품명과 유통기한 인식 전송

Technical Background – OCR

사용 이유: 수동 입력 대신 '자동 추출'을 통해 사용자의 편리함

*'Tesseract' 사용 이유:

- LSTM 기반의 모델로 높은 인식 정확도
- 100개 이상의 언어 지원 (한글 포함)
- 무료 소프트웨어
- 훈련 가능(Google Cloud Vision, Amazon Textract는 지원 X)



Problem & Plan- OCR

1. 카메라 설치 위치

(1) 냉장고 내부 => 캡처-인식 속도 차이를 고려해 프레임 캡처 간격을 길게 설정 가능, but 이전 인식된 제품들과의 구분 문제

(2) 냉장고 외부 => 장 본 재료를 정리할 때 카메라에 보여주는 식으로

한 번에 하나의 제품, 유통기한 인식 가능, but 프레임 캡처 간격의 문제

2. 프레임 캡처 속도와 tesseract 인식 속도의 차이, spring 서버로의 전송 속도를 고려하여 프레임 캡처 간격 결정

3. Tesseract 한글 인식률 확인

=> 한글 인식률 높일 수 있는 전처리 방법 선택(어떤 크기로 자르는 것이 좋을지, 그레이 스케일 변환, 이진화, 노이즈 제거 등)

4. 브랜드 별 폰트에 따른 인식률에 편차를 보일 것이라 생각함

=> (1) 다양한 실험을 통해 평균적으로 높은 정확도를 보이는 전처리 방법 선택 예정

=> (2) 너무 낮은 인식률을 보일 경우 제품 사진을 활용하여 훈련

5. 유통기한 폰트 크기의 문제 => 전처리 방법 중 팽창 사용해서 두께를 증가시켜볼 예정

03 What will be done & by whom next week?

- **Add/Remove/Fix the features of the app** @all
- **Designing the prototype of app with Figma** @진서영
- **Fixing API documentation** @황수영 @최진욱 @김민수
- **Backend** @황수영
- **Frontend** @최진욱 -> 디자인 나오는 대로 작업 진행
- **OCR Feature Implementation** @조서영 -> 카메라 도착 이후, 환경 세팅 및 프레임 캡처까지 진행 예정