

Mid Presentation

Team 1 아가나다



황선영
윤종선
김현수
박재형

CONTENTS



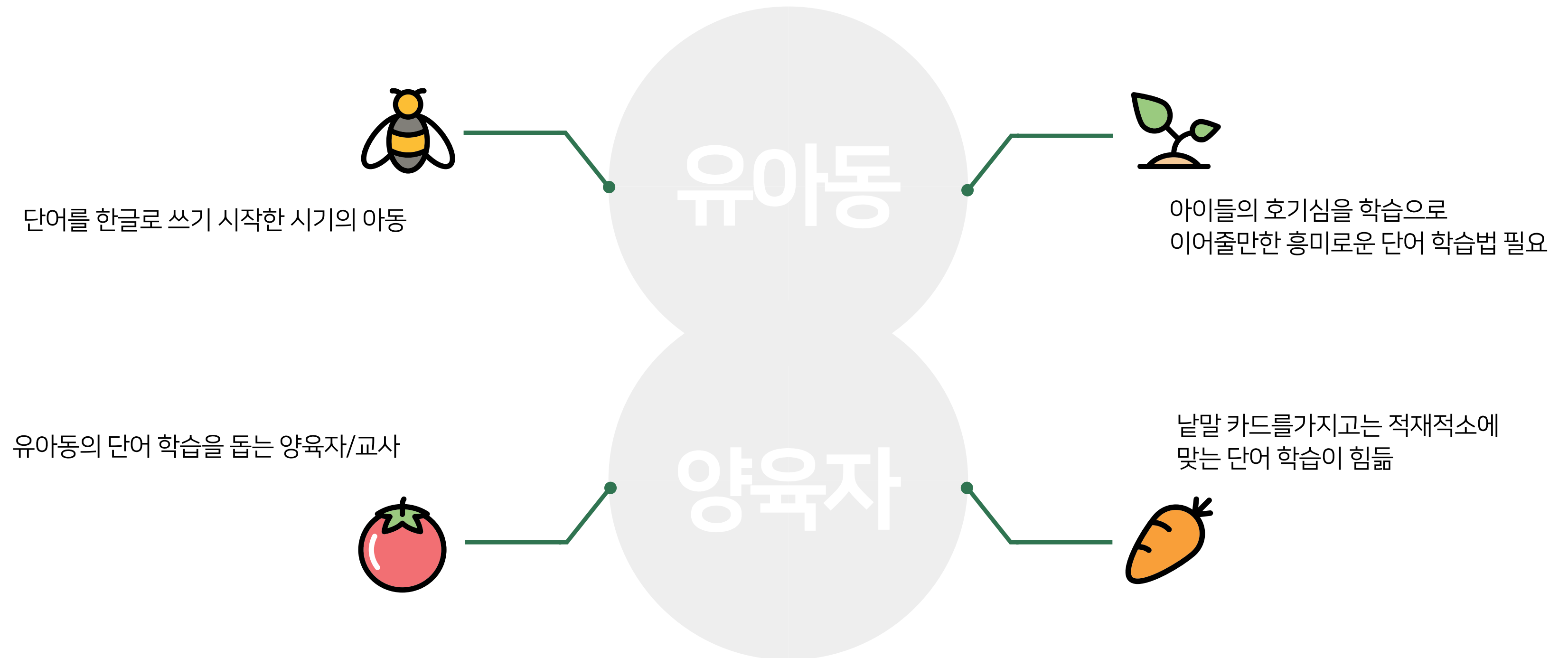
- 01 Refined project idea
- 02 Use cases & demonstration of key functions
- 03 The system architecture overview
- 04 Key technical challenges & solutions
- 05 Project schedule
- 06 Final deliverable and success criteria

01

Refined project idea



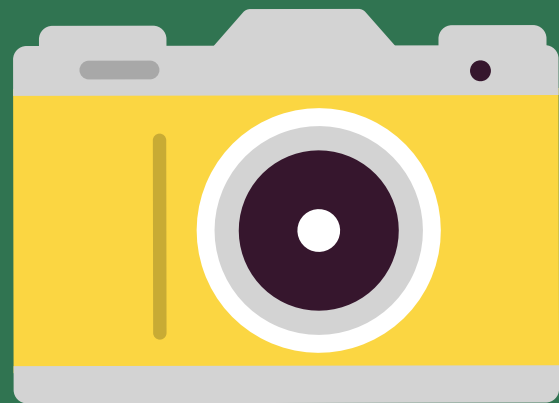
Target users and problems they are facing



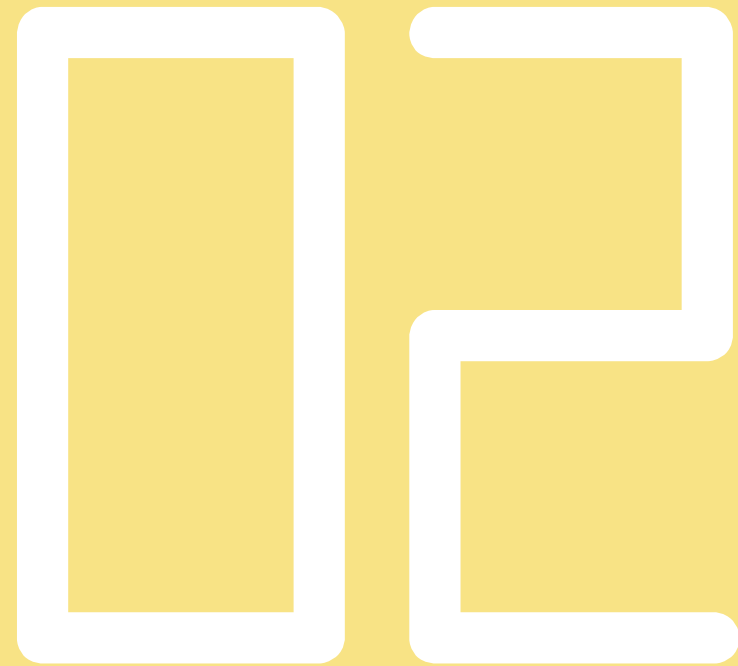
SOLUTION

아가나다

사진을 찍어
직접 학습할 단어 선택



- ✓ Real World와의 직접적인 interaction
 - > 흥미와 호기심 증가
 - > 학습으로 이어짐
 - > 학습자립심 증진
- ✓ 주변에 있는 모든 물체를 학습할 수 있음
 - > 넓은 학습 범위



Use cases & demonstration of key functions

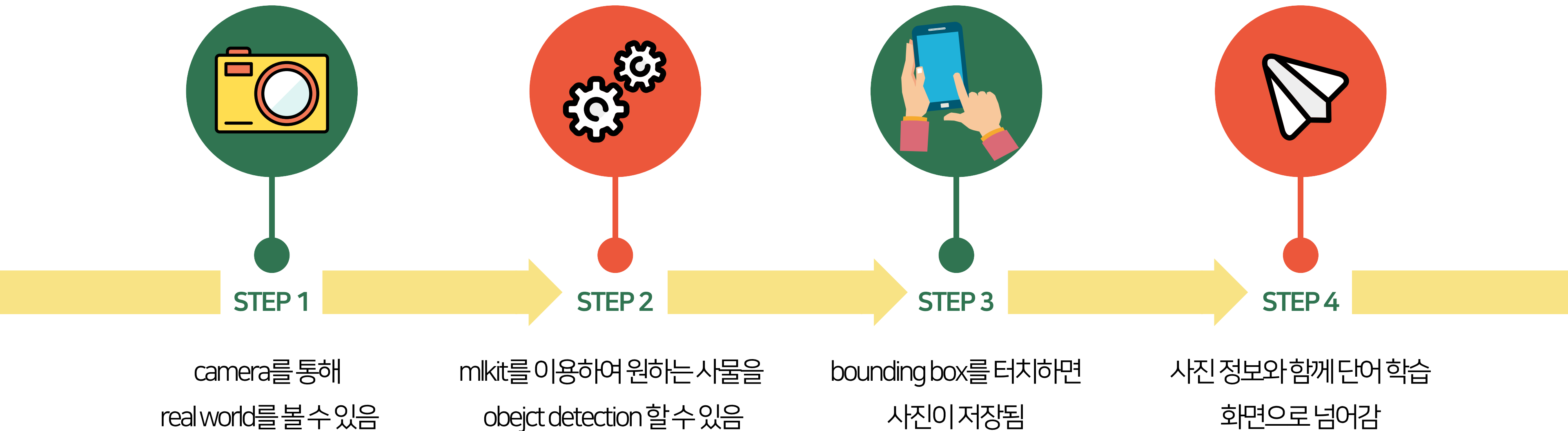




Use cases 단일 케이스

한글 쓰기 연습을 위해 주변 사물의 사진을 찍어
직접 원하는 사물을 선택하고 한글 쓰는 법을 익혀나감

Key function - 주변 사물 인식



Key function - 글자 인식 (학습/테스트)



STEP 1

사진 정보를 받아
사진과 단어를 띄움



STEP 2

글자를 따라 쓰도록 유도



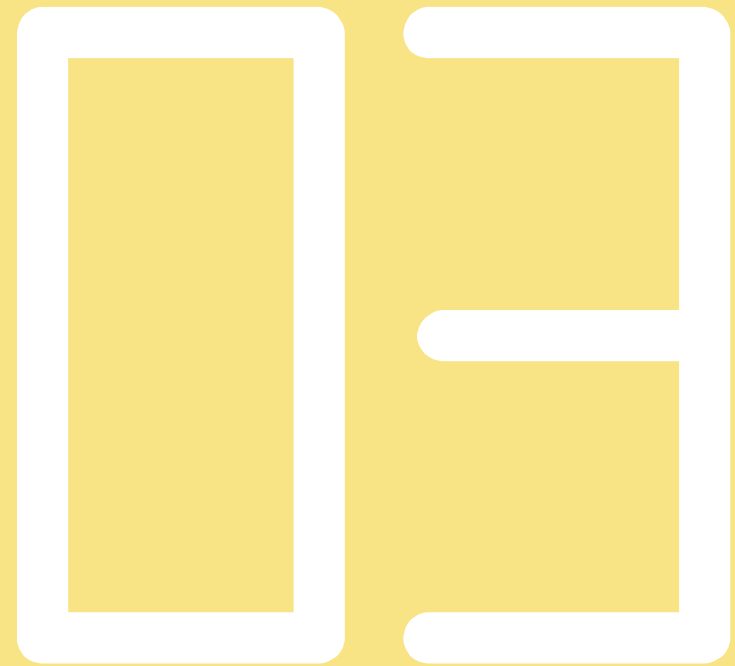
STEP 3

글자를 인식하여 채점

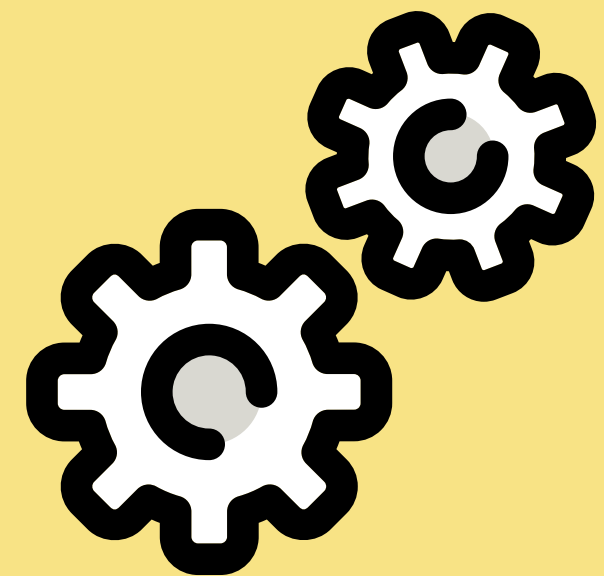


STEP 4

정답이면 칭찬을,
오답이면 응원을 보냄



The *system* architecture overview



System architecture

Description

01 Room DB를 사용하지 않고,
file system을 이용하여 data를 저장

02 MVVM architecture 사용
(model은 file system으로 대체)

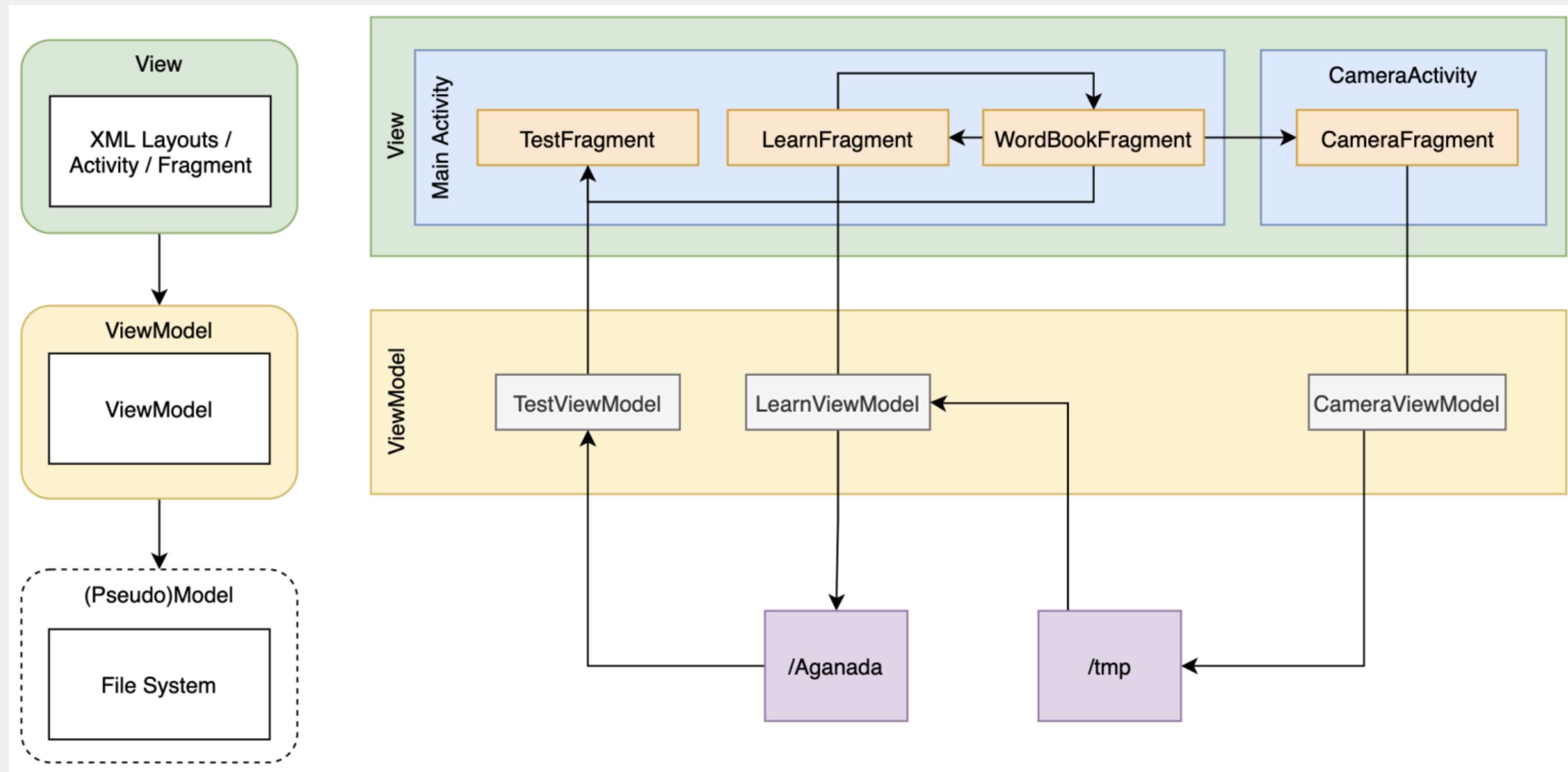
FileSystem hierarchy

com.example.aganada

tmp

Aganada

System architecture - Schema



04

Key technical challenges & solutions



Key technical challenges & solutions - 선영

Challenge

1. mlkit의 object detection model이 classify하는 범위가 컸음
2. local database를 쓰지 않고도 사진과 단어를 짝지어 저장하고자 함

Solution

1. 총 600개의 사물을 구체적으로 구분하는 TensorFlow Hub의 model로 교체
2. 물체를 인식하여 찍힌 사진의 이름을 인식 결과로 저장

Key technical challenges & solutions - 현수

Challenge

1. camerax에서 AnalysisUseCase에서 얻은 결과를 이용해서 이미지 저장하는 API가 없음
2. AnalysisUseCase에서 analyzer는 ImageProxy 파일을 받음. 근데 이 이미지를 저장하기 위해서 bitmap을 생성해줘야하는데 ImageProxy 내부 함수중에 자동으로 BitMap 만들어주는 함수가 없었음
3. Analyze에서 detect한 object의 boundingBox(Rect)을 받아서 이 Rect 내부를 클릭하면 boundingBox만큼 crop해서 저장해야함.

Solution

1. 직접 bitmap 이용해서 이미지를 저장
2. 직접 bitMap 만들
3. crop 기능이 ImageProxy에 없음 ⇒ BitMap을 이용하는 함수로 직접 구현
Rect로 받은 coordinates은 실제 이미지 preview에서의 pixel값이어서 이를 View coordinate system으로 바꿔줄 필요가 있었음
이 함수는 GraphicOverlay에 있는데 버전 문제 때문에 직접 사용하지 못해서 mock functions을 만들어서 해결

Key technical challenges & solutions - 종선

Challenge

object detection model의 label map이 영어로 되어 있으므로 한국어 labeling 필요

Solution

라벨이 631개가 넘어 일일이 수작업으로 하기엔 많아서 naver의 papago openapi를 이용해 1차로 자동 번역을 거치고 잘못 번역된 단어들을 수작업으로 수정

Key technical challenges & solutions - 재형

Challenge

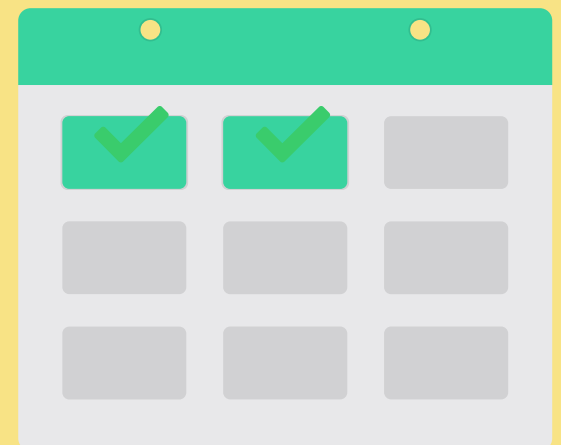
1. 사진과 텍스트가 주어졌을 때, 클릭 시 뒤집을 수 있는 단어 카드를 만들어야함
2. 단어 카드 배치

Solution

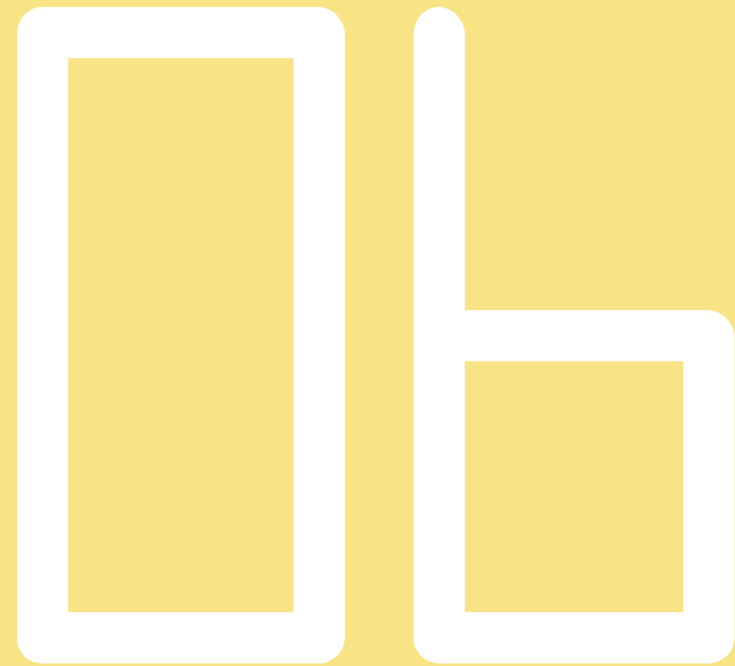
1. TextView와 ImageView 둘 중 하나를 90도 돌려놓고, 하나가 90도 돌아가면 나머지 하나도 90도 돌아가는 형식으로 구현 ⇒ 두 View는 FrameLayout 으로 합침
2. GridLayout을 사용하여 배치. 화면을 넘어갈 수 있기 때문에 ScrollView 사용

05

Project schedule



기능	개발 자	세부기능	sprint 1	sprint 2	sprint 3	sprint 4	sprint 5
시작 화면	선영	Splash 화면 띄우기/font 적용			*		
사진 촬영 화 면	선영	CameraX를 이용해 카메라 기능 구현	*				
	선영	MLkit object detection 반영	*				
	현수	Detect된 object 정보 이용해서 사진 크 기 조정		*			
	현수	Detect된 object의 이름으로 사진 저장		*			
단어 학습 화 면	종선	글자 쓰기 기능	*				
		사진촬영구현되면 사진/단어 정보 가져오 기			*		
		잘못된 정보 수정 기능				*	
테스트 화면	종선	테스트하기 기능 (글자 인식)			*		
채점 화면	종선	글자 인식 기능	*				
		사진촬영구현되면 사진/단어 정보 가져오 기			*		
		잘못된 정보 수정 기능				*	
단어장 화면	재형	사진 띄우기 기능	*				
		사진 뒤집기 기능	*				
		사진촬영구현되면 사진/단어 정보 가져오 기			*		
		사진 단어 수정 제거				*	
최적화	모두	디자인, 성능, 정확도, 사용감 개선				*	*



Final deliverable and success criteria



Final deliverable

1. 사진을 찍고, 해당 물체의 단어를 찾고, 따라 쓴 글씨가 정상적인지 판별한다.
2. 학습한 내역을 물체사진-단어로 저장하여 이를 단어장에서 확인할 수 있다.
3. 물체사진-단어로 간단한 test 를 제공할 수 있다.

Success criteria

1. 사진을 찍고 물체를 detect 하여 물체를 선택할 수 있다.
2. 물체에 해당하는 한글 단어를 찾을 수 있다
3. 단어를 따라 쓸 수 있다.
4. 따라 쓴 글씨가 정상적인지 판별할 수 있다.
5. 물체사진-단어 가 잘 저장되고 이를 확인할 수 있다.



1

2

3

4

Thank you

Team1 아가나다

