

Smart K-art

OCR pytesseract를 활용한 똑똑한 쇼핑 도움 서비스



<옥장판>

박지영
성채원
정성윤

이미지분석기반의 인공지능 플랫폼 개발자 양성과정

목차

01 서비스 개요

02 서비스 구성

03 프로젝트 관리

04 서비스 발전 방향





Part 1.

서비스 개요

1. 서비스 개요

서비스 명

- Smart K-art (똑똑한 K-장바구니)

서비스 개요

- OCR 기반 계산형 장바구니 서비스

프로젝트 기간

- 2020.09 ~ 2020.11.21 (3개월)

팀원

- 박지영, 성채원, 정성윤
(이미지 분석기반 인공지능 플랫폼 개발자 양성과정 2분반 교육생)

멘토

- 김동억 기술사, 조숙향 기술사

2. 기획 배경 및 의도

001 > 기존의 마트 장보기 관련 기능의 개인화

002 > 쇼핑현장에서 충동구매로 인한 과소비 방지

003 > 개인의 장보기 습관을 시각적으로 확인

2. 기획 배경 및 의도

001 > 기존의 마트 장보기 관련 기능의 개인화



Amazon Dash Cart: 카트 속 자동결제 솔루션



롯데마트: 오프라인매장 전 상품에 배송을 위한 QR코드 도입

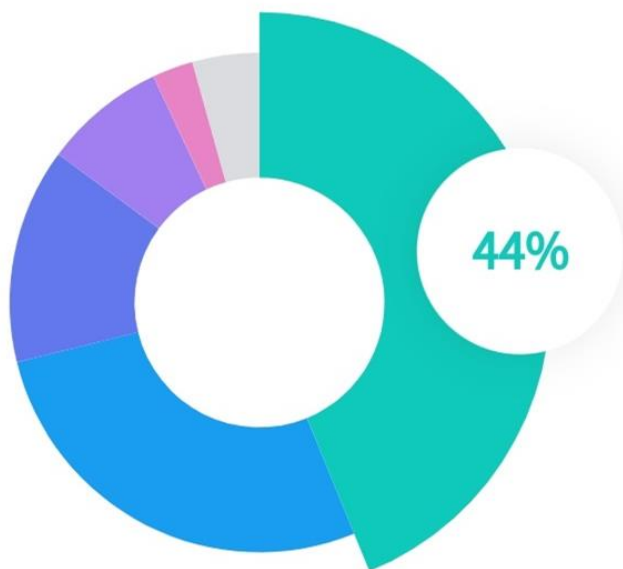
[Smart K-art의 차별점]

마트마다 다른 도구와 개발이 필요한 기존 쇼핑 편의 기능과 달리,
개인 휴대폰으로 어디서든 제약없이 사용 가능

2. 기획 배경 및 의도

003 > 개인의 장보기 습관을 시각적으로 확인

◎ 지출 445,720원



전체

445,720원

● 식재료 44%

195,300원 >

● 위생용품 27%

121,800원 >

● 유제품 14%

62,250원 >

● 마스크 8%

35,260원 >

● 뷰티/미용 3%

11,820원 >

● 생수 2%

10,000원 >

● 주류 2%

9,290원 >

3. 페르소나(Persona)

페르소나 > 서비스를 사용할 가상의 사용자로서, 서비스에 대한 사용자의 생각이나 필요성 등 예측



- 이름: 김지혜
- 나이: 28
- 성별: 여
- 직업: 일반 회사원
- 거주지: 서울시 노원구 주거지역
- 취미: 집밥 요리하고 SNS에 인증하기
- 휴대폰 기종: IPHONE 11

특징

- 1인 가구
- 주 3회 이상 직접 조리한 음식으로 식사
- 2주에 한 번 식료품 구매를 위해 대형마트 방문
- 평소 휴대폰 하나만 소지하고 활동

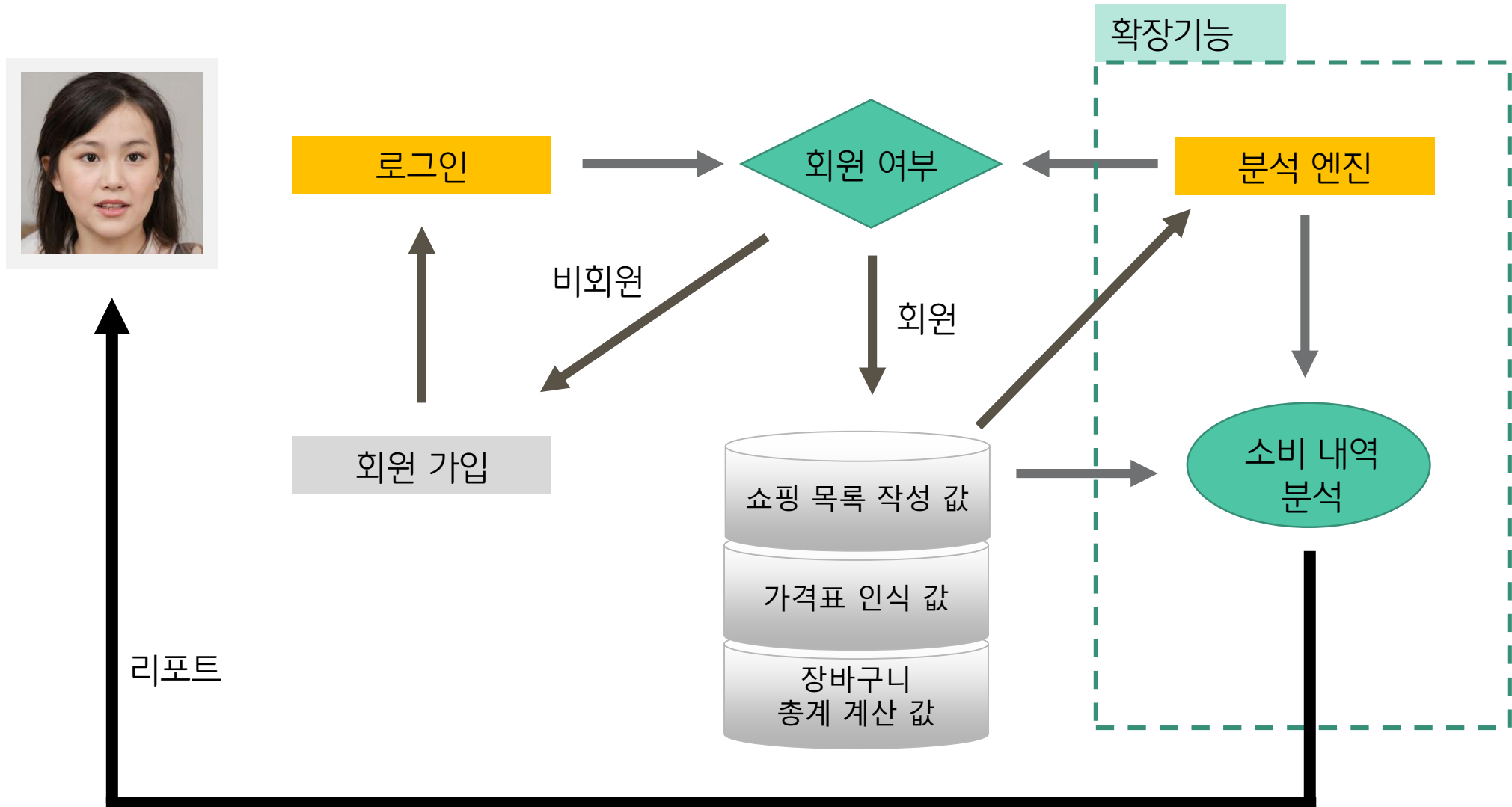
NEEDS

- 마트 장보기 전 예산과 실제 장본 금액 간 차이 줄이기
- 쇼핑 목록과 실제 장바구니 목록 비교 통한 충동구매 방지 필요
- 휴대폰 하나만 있으면 이용가능하도록 시공간 제약 없는 환경 필요

4. 서비스 시나리오

1	사용자가 평소 필요하다고 생각한 제품을 SMART K-ART 앱의 쇼핑 목록에 작성
2	식료품 구매를 위해 집근처 대형마트 방문
3	마트에서 SMART K-ART 앱의 '카메라'를 클릭
4	제품의 가격표를 찍고, 해당 이미지를 OCR 인식을 통해 제품명, 가격 추출(반복 동작)
5	장바구니 목록에 추가된 제품 상세보기에서 수정 및 삭제
6	구매예정 목록과 장바구니 목록을 한눈에 보고, 예산과 실제 구매금액 비교를 통한 충동구매 및 과소비 방지
7	쇼핑 완료한 해당 목록은 지난 쇼핑목록으로 이동 후, 이전의 구매 목록 확인을 통해 중복 구매 방지 및 구매 패턴 확인

5. 서비스 흐름도

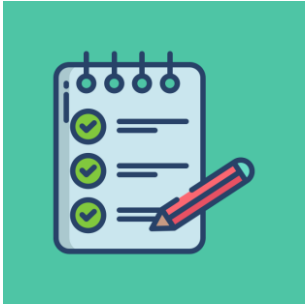


A person wearing a white lab coat and dark shoes is walking on a light-colored concrete sidewalk. A long, dark shadow is cast on the ground to the left of the person. The person is carrying a black bag. The background is a solid light blue color.

Part 2.

서비스 구성

1. 주요기능



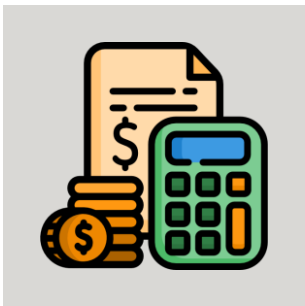
1. 쇼핑 목록 작성 기능

- 과소비를 방지하기 위한 사전 쇼핑 목록 작성
- 작성한 쇼핑 목록과 현재 장바구니 비교



2. 가격표 인식 기능

- 매대의 가격표를 인식하여 품명과 가격 자동 입력



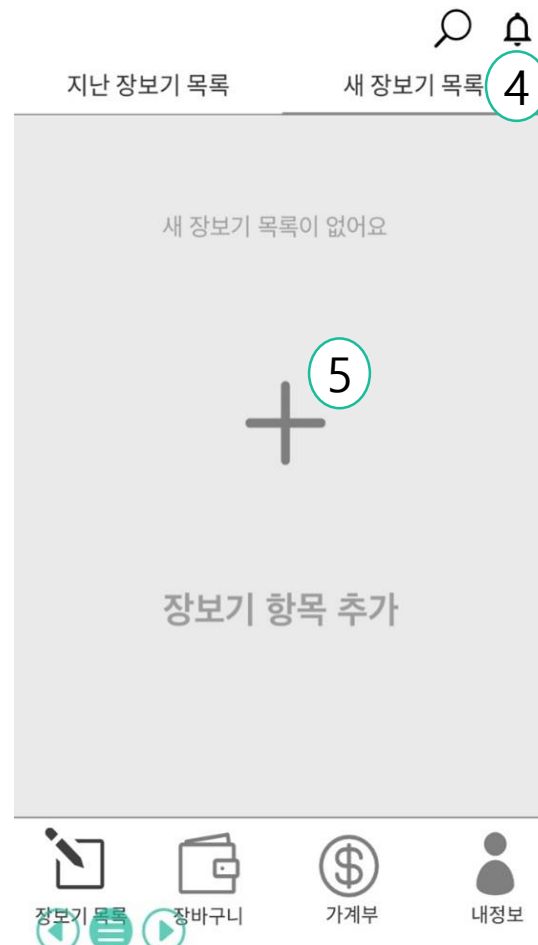
3. 장바구니 총계 계산 기능

- 현재 장바구니에 들어있는 물품 총계 계산
- 지난 쇼핑목록으로 가계부 역할 연동
- 추후 구매패턴 분석용 데이터 수집

2. 화면 구성 - UI 설계도



홈화면

장보기 목록 -
지난 장보기 목록장보기 목록 -
새 장보기 목록

- (1) 장보기 목록 클릭
-지난 장보기 목록 화면 출력
- (2) 제품 추가 클릭
-가격표 찍을 카메라 연결
- (3) 지난 장보기 목록 클릭
-과거 장본 목록(날짜, 이름 등) 출력
- (4) 새 장보기 목록 클릭
-장보기 항목 추가 화면 출력
- (5) 장보기 항목 추가 클릭
-새 장보기 목록2로 연결

2. 화면 구성 - UI 설계도

The four screens illustrate the following states and actions:

- Screen 6:** Initial state for adding a new item. Fields for 'Topic' (이마트 3), 'Add Item' (제품, 수량), and 'Add Item' (추가) button.
- Screen 7:** State after adding items. A table shows added items (No., Name, Value) with checkboxes for selection. Buttons for 'Select All' (전체 선택), 'Deselect All' (선택 삭제), and 'Complete' (작성 완료) are visible.
- Screen 11:** Confirmation dialog for adding a new item to the list. Text: '이마트 3 으로 장보기 목록을 추가하시겠습니까?' (Do you want to add the shopping list to I-Mart 3?). Buttons: '아니요' (No), '네' (Yes).
- Screen 12:** State after selecting items. A table shows selected items (제품, 수량, 가격) with checkboxes. Buttons for 'Select All' (전체 선택), 'Deselect All' (선택 삭제), and 'Delete All' (전체 삭제) are visible.
- Screen 13:** Confirmation dialog for deleting items. Text: '이마트 3 장보기를 완료 처리 하시겠습니까?' (Do you want to complete the I-Mart 3 shopping list?). Buttons: '아니요' (No), '네' (Yes).
- Screen 14:** State after deleting items. A button labeled '장보기 완료' (Shopping List Complete) is visible.
- Screen 15:** Confirmation dialog for completing the list. Text: '이마트 3 장보기를 완료 처리 하시겠습니까?' (Do you want to complete the I-Mart 3 shopping list?). Buttons: '아니요' (No), '네' (Yes).

새 장보기목록2

새 장보기목록2
- 추가 질문

새 장보기목록2
- 선택 및 삭제

새 장보기목록2
- 완료 질문

- (6) 추가 버튼 클릭
-추가할 제품 명, 수량 추가
-추가된 항목으로 출력
- (7) 추가 항목 체크박스 클릭
-선택된 제품 확인
- (8) 선택 삭제 버튼 클릭
-체크박스에 선택된 항목 삭제
- (9) 전체 선택 버튼 클릭
-추가된 항목 체크박스 전체 선택
- (10) 작성완료 버튼 클릭
-추가 질문으로 연결
- (11) 추가질문
-제목, 추가 제품 등을 장보기 목록에 추가 여부
- (12) 새 장보기 목록 화면
-이전 수정 사항 반영
- (13) 체크박스 선택 및 삭제
-전체 및 선택 항목 삭제
- (14) 장보기 완료
-완료 질문으로 연결
- (15) 완료 질문
-해당 장보기 목록 완료 여부

2. 화면 구성 - UI 설계도



(16) 장바구니 클릭
-현재 장바구니 화면 출력

(17) 드롭다운 박스 클릭
-최근 목록을 default
-장보기 목록에 저장된 리스트
확인 및 클릭

(18) 가계부
-결제 내역(일자,금액,목록)
리포트
-결제 분석 내역 리포트

(19) 내정보
-사용자 프로필 확인
-자동 결제 카드 설정
-자주 찾는 매장 설정
-친구초대
-공지사항
-자주 묻는 질문
-앱 설정

3. CRUD Matrix

엔티티 집합 단위 프로세스	회원	쇼핑목록	지난 쇼핑목록	장바구니
로그인	R			
회원가입	C			
회원정보 수정	U			
회원정보 삭제	D			
쇼핑목록 입력		C		
쇼핑목록 조회		R	R	
쇼핑목록 수정		U	U	
쇼핑목록 삭제		D	D	
장바구니 입력				C
장바구니 조회				R
장바구니 수정				U
장바구니 삭제				D

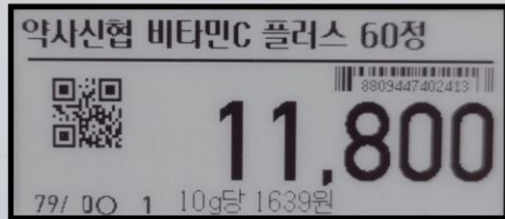
3. CRUD Matrix

Create

가격표:

파일 선택 선택된 파일 없음

이미지 업로드



약사신협비타민플러스60정

11800

저장

다시찍기

Read

장바구니 리스트

카메라

쇼핑 계획

작성일	2020년 11월 23일	목적	기본
제목	집들이	예산	50000
내용	파스타소스 휴지 과자 맥주		

장바구니

제품명	수량	가격
약사신협비타민플러스60정	1	11800

총계: 11800원

Update

Smart K-art

쇼핑목록 지난쇼핑목록 내정보 로그아웃

안녕하세요 jsyoon777@naver.com 님

수정

번호	65
수량	1
제품명	약사신협비타민플러스60정
가격	11800

저장

Delete

Smart K-art

쇼핑목록 지난쇼핑목록 내정보 로그아웃

안녕하세요 jsyoon777@naver.com 님

조회

번호	65	수량	1
제품명	약사신협비타민플러스60정		
가격	11800		

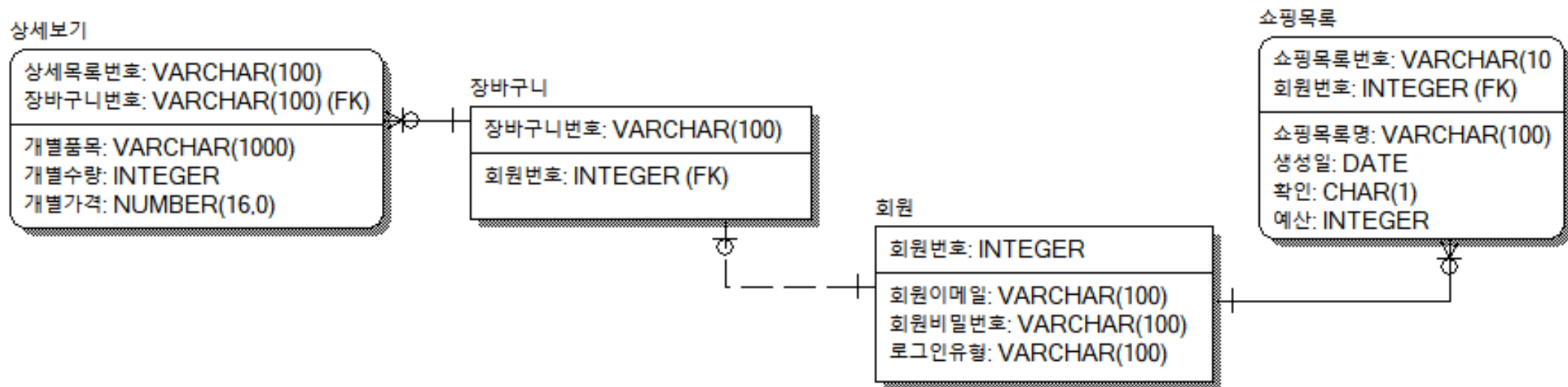
수정

삭제

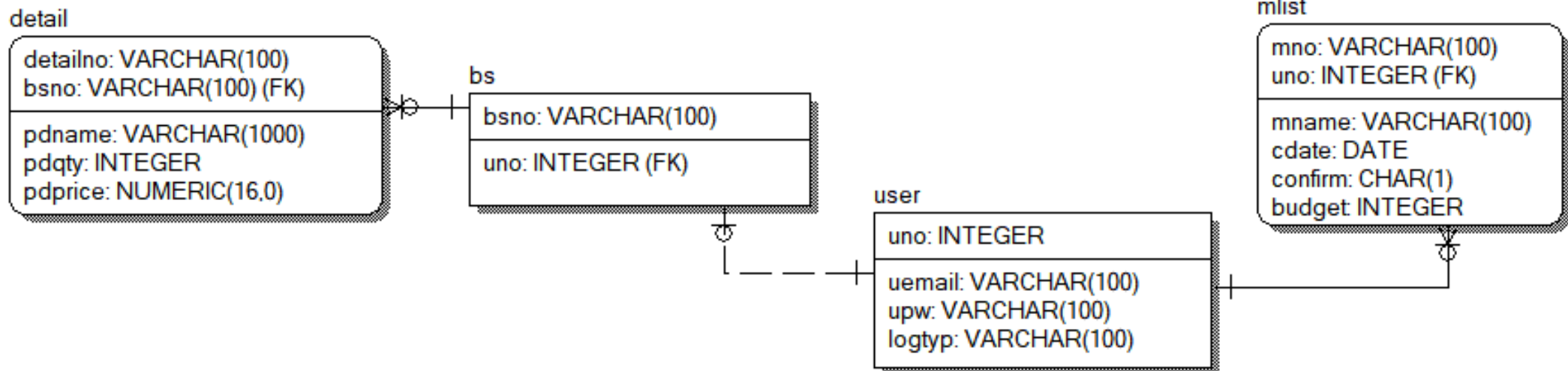
리스트

4. 데이터 모델 - ERD

논리 모델



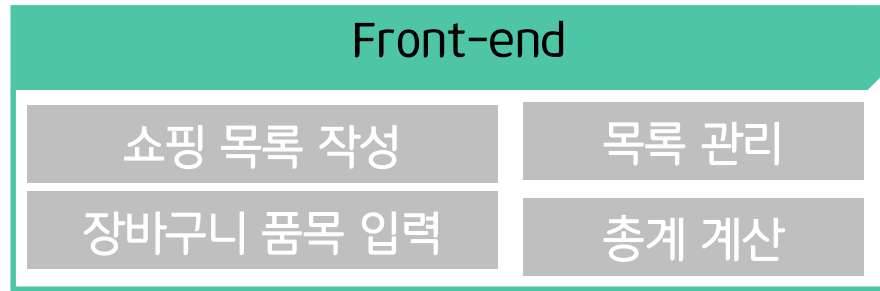
물리 모델



4. 데이터 모델 - 테이블 명세서

Logical	full-name	Physical	Data Type	Null	Default
회원	user	user			
회원번호	userNumber	uno	VARCHAR(100)	Not Null	
회원이메일	userEmail	uemail	VARCHAR(100)		
회원비밀번호	password	upw	VARCHAR(100)		
로그인유형	loginType	logtyp	VARCHAR(100)		
쇼핑목록	memoList	mlist			
쇼핑목록번호	memoNumber	mno	VARCHAR(100)	Not Null	
쇼핑목록명	memoName	mname	VARCHAR(100)		
생성일	createDate	cdate	DATE		
확인	confirm	confirm	CHAR(1)	Not Null	N
예산	budget	budget	INTEGER		
상세보기	detail	detail			
상세목록번호	detailNumber	detailno	VARCHAR(100)	Not Null	
장바구니	basket	bs			
장바구니번호	basketNumber	bsno	VARCHAR(100)	Not Null	
개별품목	productName	pdname	VARCHAR(1000)		
개별수량	productQuantity	pdqty	INTEGER		
개별가격	productPrice	pdprice	NUMBER(16,0)		

5. 시스템 아키텍처



수집 엔진
- Python



처리/분석 엔진

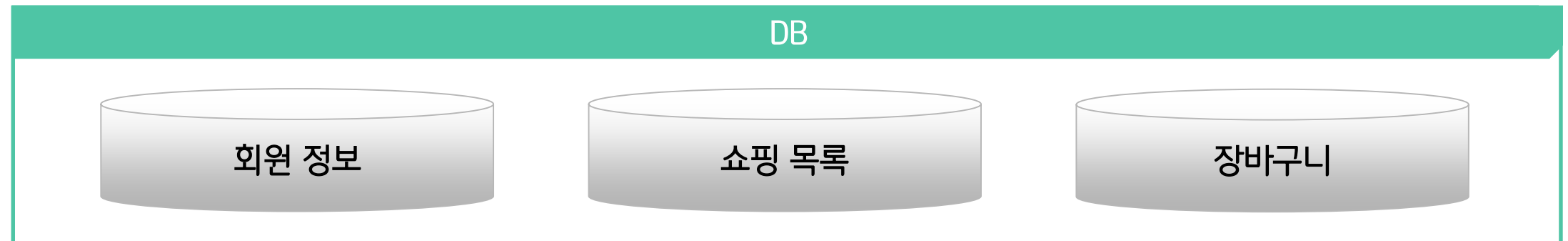
- Python
- OCR
- Tesseract
- Numpy
- OpenCV



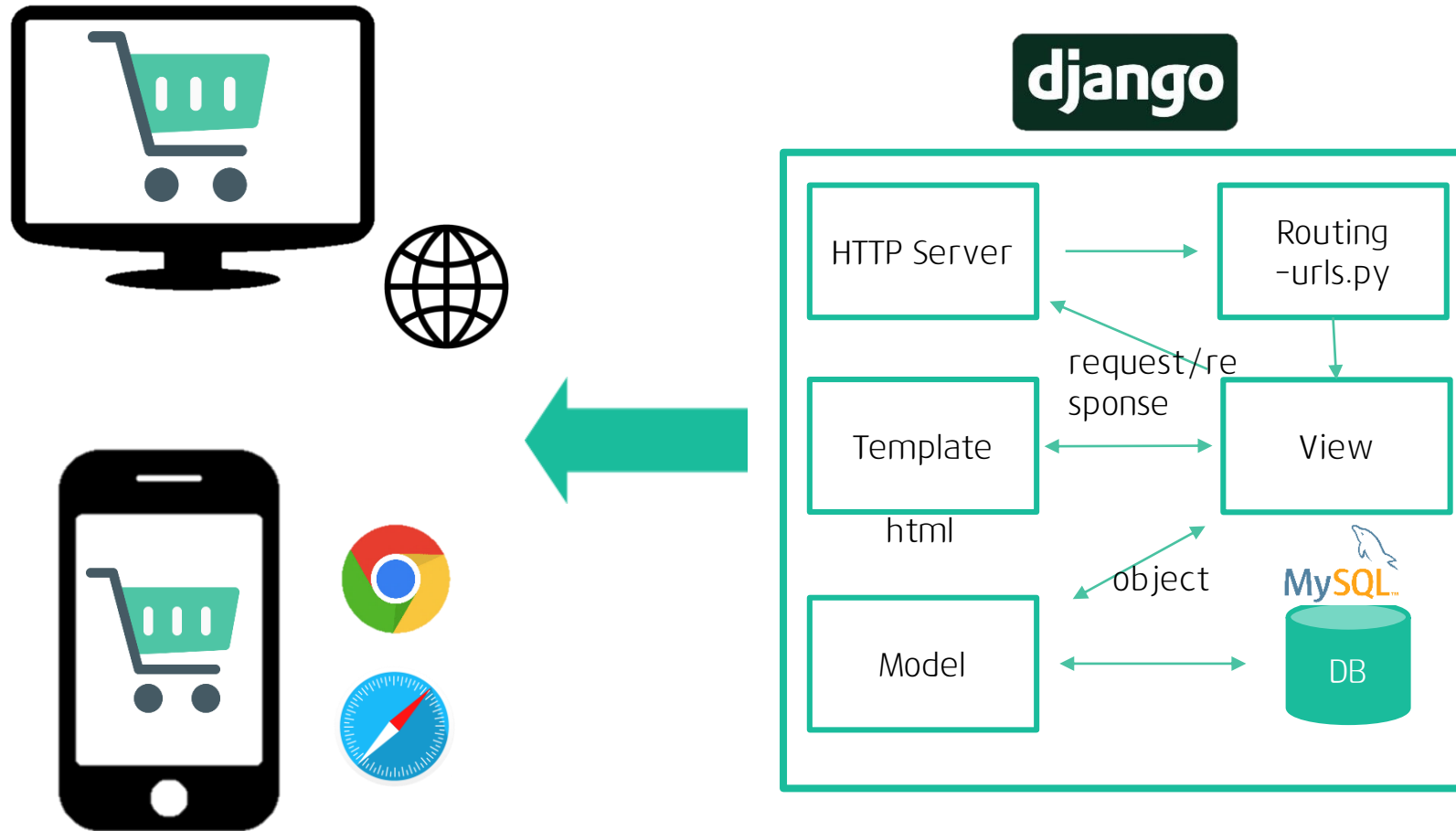
추출/저장 엔진
- Python
- MySQL



출력 엔진
- django

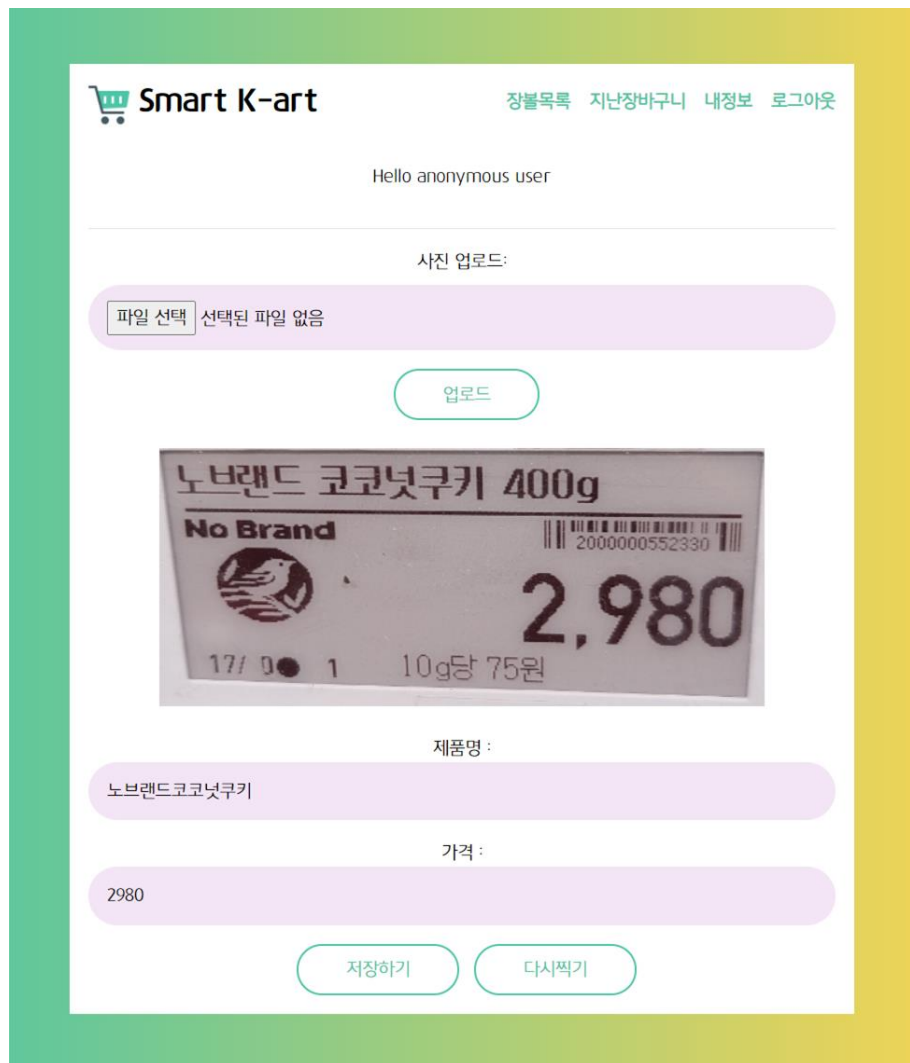


6. 웹 서비스 아키텍처



Python 기반
웹프레임워크
Django를 통한
웹 개발 및
MVT 모델 구현

7. OCR을 활용한 데이터 처리



The screenshot shows the Smart K-art web application interface. At the top, there's a header with the logo and navigation links. Below the header, a message "Hello anonymous user" is displayed. The main section is titled "사진 업로드:" (Photo Upload:). It features a file selection area with a button "파일 선택" (File Select) and a status "선택된 파일 없음" (No file selected). Below this is an "업로드" (Upload) button. The uploaded image is a product label for "노브랜드 코코넛쿠키 400g" (No Brand Coconut Cookie 400g) with a price of "2,980". The label also includes a barcode and the text "10g당 75원". Below the image, the application displays the extracted data: "제품명:" (Product Name) as "노브랜드코코넛쿠키" and "가격:" (Price) as "2980". At the bottom, there are two buttons: "저장하기" (Save) and "다시찍기" (Retake).

OCR(광학 문자 인식)

사람이 쓰거나 기계로 인쇄한 문자의 영상을
이미지 스캐너로 획득하여 기계가 읽을 수 있는
문자로 변환하는 기술

SMART K-art의 데이터 처리

1. 가격표 이미지 인식 (OpenCV)
2. 문자인식(OCR)
3. 품명과 가격을 자동으로 기입
4. 데이터 저장

7. OCR을 활용한 데이터 처리

1. 가격표 이미지 인식(OpenCV - ROI)
2. 문자 인식(OCR)
3. 인식된 데이터 전처리
 - 특수문자, 공백 등 불용어 제거
 - 추출 텍스트 리스트화
 - 추출 리스트의 품명과 가격을 인덱스로 설정하여 자동으로 기입
 - 사용자 수정 데이터 호출
 - 수정된 가격 문자열에서 정수로 변경
4. 데이터 저장

ROI (관심 영역 검출)

```
src=cv2.imread('C:/deep/opencv/image/unnamed4.jpg')

dw, dh= 720, 400
srcQuad=np.array([[0,0],[0,0],[0,0],[0,0]],np.float32)
dstQuad=np.array([[0,0],[0,0],[0,0],[0,0]],np.float32)
dst=np.zeros((dh,dw), np.uint8)

src_gray=cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)
th, src_bin=cv2.threshold(src_gray, 255, 255, cv2.THRESH_BINARY)
#외곽선 검출
contours, _ =cv2.findContours(src_bin, cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)

for pts in contours:
    #너무 작은 객체 제외
    if cv2.contourArea(pts) < 100:
        continue
    #외곽선 근사화
    approx=cv2.approxPolyDP(pts, cv2.arcLength(pts, True)*0.02, True)
    #컨벡스가 아니면 제외
    if not cv2.isContourConvex(approx) or len(approx) != 4:
        continue
    srcQuad=reorderPts(approx.reshape(4, 2).astype(np.float32))
    pers=cv2.getPerspectiveTransform(srcQuad, dstQuad)
    dst=cv2.warpPerspective(src, pers, (dw, dh), flags=cv2.INTER_CUBIC)
    dst_rgb=cv2.cvtColor(dst, cv2.COLOR_BGR2RGB)
    print(pyesseract.image_to_string(dst_rgb, lang='Hangul+eng+kor'))

cv2.imshow('src', src)
cv2.imshow('dst_rgb', dst_rgb)
cv2.waitKey()
cv2.destroyAllWindows()
```



7. OCR 엔진 - Tesseract 학습

Tesseract : 다양한 OS를 지원하기 위한 OCR 엔진, Apache 라이선스

⇒ Tesseract 4 버전부터 LSTM 기반의 OCR 엔진 추가

	Trained models	Speed	Accuracy	Supports legacy	Retrainable
tessdata	Legacy + LSTM (integerized tessdata-best)	Faster than tessdata-best	Slightly less accurate than tessdata-best	Yes	No
tessdata-best	LSTM only (based on langdata)	Slowest	Most accurate	No	Yes
tessdata-fast	Integerized LSTM of a smaller network than tessdata-best	Fastest	Least accurate	No	No

Tessdata-best : 속도는 느리지만 가장 정확하고 학습이 가능

⇒가장 성능이 좋음

7. OCR 엔진 - Tesseract 학습

Tessdata



노브랜드코코넛쿠키4000

22980

저장

다시찍기

Tessdata-best



노브랜드코코넛쿠키400

2980

저장

다시찍기

Tessdata-fast



노브랜드4000

1

저장

다시찍기

7. OCR 엔진 - Tesseract 학습

Tessdata



041

프

저장

다시찍기

Tessdata-best



갈릭화이트조스4109

4.980

저장

다시찍기

Tessdata-fast



갈릭화이트그410

4.980

저장

다시찍기

8. 주요 데이터 처리 프로세스

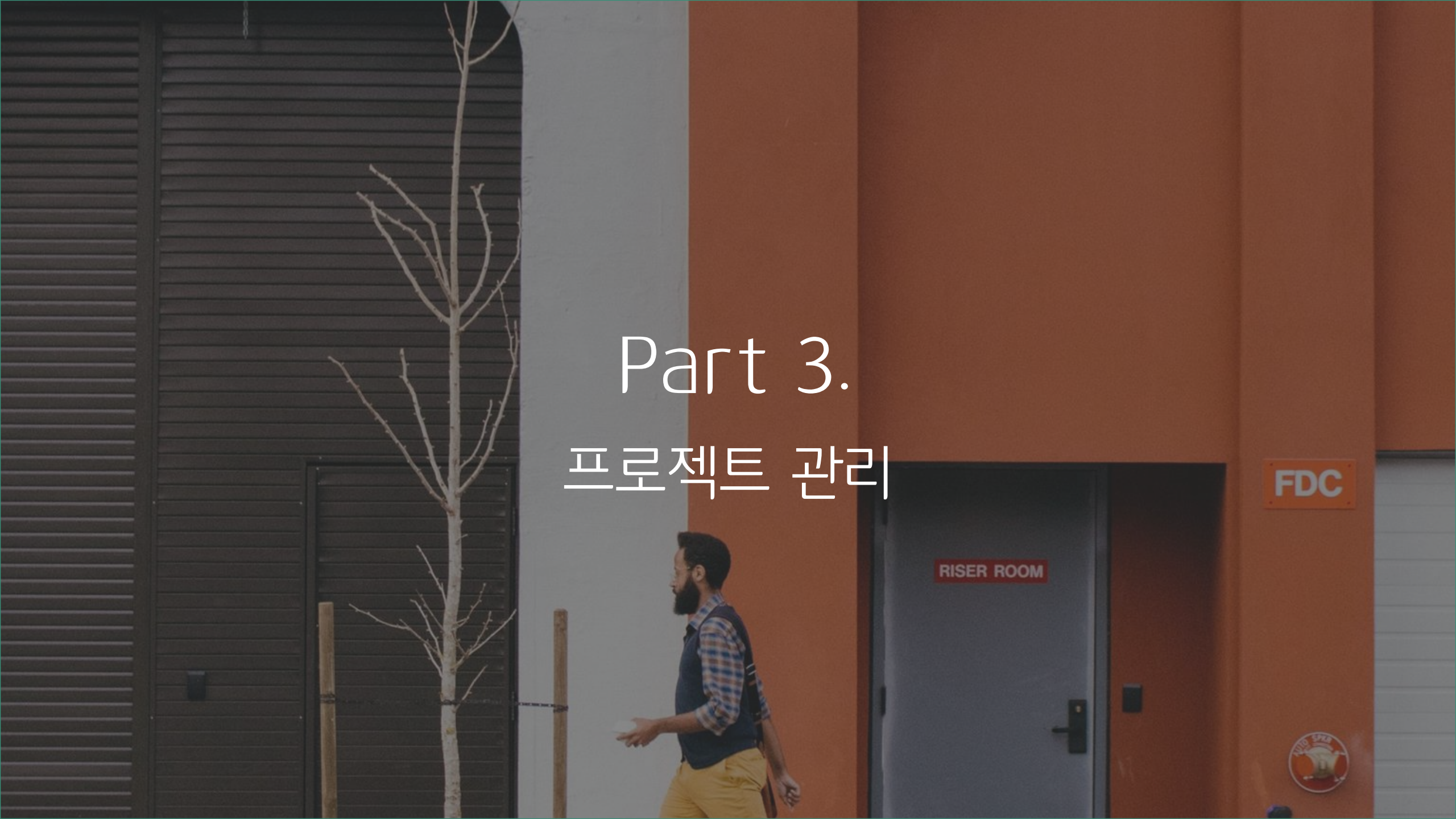


웹 서비스 작동 시연

49.247.22.191:8000

Part 3.

프로젝트 관리










1. Smart K-art 프로젝트 개발 일정

[illegible]

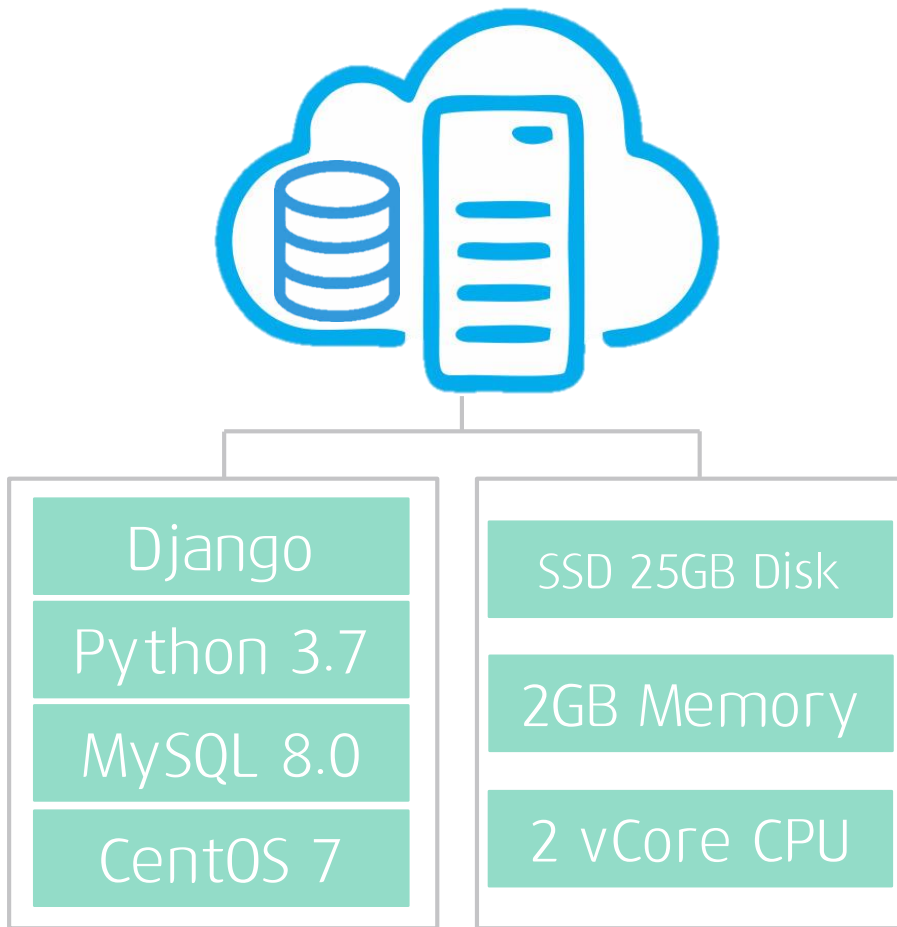
2. 팀 구성 및 업무분장

#	분류	TASK	담당자
1	서비스 기획	서비스 기획 및 기획서 작성 UI 설계 및 작성	성채원 정성윤
2	Server	Linux 설치 관리자 권한 관리	성채원
3	Backend	Django MVT 모델 설계/ 구현 MySQL-Django ORM 모델 설계/구현	박지영 성채원
4	DB	Server DB- MySQL 연동 논리/물리 데이터베이스 모델 설계	박지영 성채원 정성윤
5	Front-end(웹)	반응형 Website UI 설계 HTML/CSS/JS 템플릿 제작	박지영 성채원 정성윤
6	이미지 인식	OCR 기능 구현 텍스트 전처리 기능 구현	박지영 성채원

3. 개발도구

	Tools		설명	framework	language
Server		MobaXterm	서버 제어 프로그램		Linux(CMD) 
		FileZilla	파일 전송 프로그램		
Front-end		Visual studio code	파이썬 통합 개발 환경		
Back-end		Visual studio code	파이썬 통합 개발 환경		Python 
		Jupyter notebook			
DB		MySQL workbench	데이터베이스 설계 프로그램		

4. 개발 및 운영 환경 구성



IP: 49.247.22.191

클라우드 서버 업체 스마일서브

Since 2002
SMILESERV

서버 스토리지 데이터베이스 백업 & 보안 콘텐츠 계정 & Biz 도메인 & DNS

구분 등록 메뉴얼 기술지원 게시판

현재판넌

서버 관리 (가상 서버)

서버 생성하기

서버 관리

목록 다운로드

새로고침

검색

검색 해제

필터 초기화

서버 수량

1

대

ZONE

이름

운영체제

공인IP

사설IP

리드타임

Status

Port

Resource

방화벽

신규일

KR1-Z05

지름선서버

CentOS

49.247.22.191

10.5.0.12

vCore.V3

운영중

off

off

off

2020-09-12 17:32:01

상세 정보

업그레이드/다운그레이드

Port 모니터링

Resource 모니터링

서버 상세



정보 수정

ZONE

KR1-Z05

Node

7c0034db71f623a58437614c9f17f0a8d79b40a2e80f515863c99bf5

이름

지름선서버(49.247.22.191) [VNC Console 접근]

IP

공인:49.247.22.191 [IP추가], 사설: 10.5.0.12

ID

3b7a0abd-cd93-4205-a04a-fac4c60ead52

운영체제

CentOS 7.X(64bit) [운영체제 재설치]

용량

서버(25 GB)

서비스

vCore.V3 [상세정보]

Status


운영중

플래그(메니지드)

미사용

신청일

2020-09-12 17:32:01



Part 4.

서비스 발전 방향

1. 서비스 발전 방향



소비 패턴 분석 기능

주기적인 사용자
소비 패턴 분석



구독 상품 자동 결제

패턴을 분석 하여
주기적으로 구매하는
상품을 구독하는 서비스



OCR 정확도 개선

DB에 저장된 가격표
이미지의 폰트를 학습시켜
OCR 정확도 개선

감사합니다

Feedback



‘시스템 아키텍처’ 부분이 인상 깊었다.

반응형 웹을 선택한 이유는:
모바일 기반 서비스이기 때문에 휴대성이 꼭 필요.
+React 기반 어플리케이션 개발은 현실적으로 어려움



‘충동구매’ : 피피티와 기능의 매치

기존 상품의 데이터베이스를 추가 활용하여 소비자가격(시세)과 비교 매칭:
현재 내가 합리적인 소비를 하고 있는 것이 맞는지 확인할 수 있도록!