

# 5/12 회의록

|      |               |
|------|---------------|
| ☰ 태그 | 팀회의           |
| 📅 날짜 | @2025년 5월 12일 |

## ▼ 회의 준비

[미리 읽어야 할 자료나 결정해야 할 사항을 적어주세요]

## 회의 안건

1. 중간보고서 다같이 검토
2. 지난 질문을 바탕으로 답변
3. 피피티 초안 확인

## 중간보고서 검토 : 무조건 중간보고서 같이 읽어볼 것 (빨간색 파트 다 노선에 작성 안 함)

1. 선행기술 파트 (특허명세서) 작성 부분 논의 필요함.

→ “사용자 이용 이력 기반 추천” 이라고 수행 계획서에 작성되어 있음. 이 부분을 보고 추천 시스템 특허를 관련 선행 기술로 가져오신 듯함

## ▼ 중간 보고서 작성 부분

### (6) 상품 가격을 고려한 상품 추천시스템 및 그 추천방법

- 서비스 소개 : 협력 필터링(Collaborative Filtering) 알고리즘을 기반으로 사용자 구매 이력을 분석해 상품을 추천하는 서비스로, 추천 시 상품 가격을 반영하여 판매자 수익을 극대화하는 전자상거래용 추천 시스템.
- 차별점 : 판매자의 이윤 극대화를 추구한다는 점에서, 사회적 가치 창출·기부에 목적인 돈쫄과 차이점이 있음

(월요일에 교수님께 꼭 여쭙보고 작성여부 결정)

- 돈쫄의 "다른 돈쫄러들이 이 가게를 돈쫄내줬어요!" 기능은 사용자의 현재 위치를 기준으로 반경 내에서 다른 사용자들이 소비 인증(영수증 인증)을 완료한 가게 리스트를 단순 노출하는 기능. 이는 추천 알고리즘 없이, 인증 데이터 기반 조회 결과를 UI에 출력하는 구조임 ( 이 부분 조원들 읽으면 지우기)

- 차별점 : 협업 필터링 알고리즘은 협업 필터링 및 가격 가중치를 연산하여 자동 추천 목록을 생성함. 반면 돈쫌은 특정 알고리즘 없이 인증된 소비 이력 데이터를 단순 조회. 추천 시스템으로 분류되지 않으며, 기술적 구현 목적이나 방식 모두 상이함.

## 2. OCR 헛갈리는 부분

저희 그럼 사업자 번호를 외부 api ( 사업자 번호 존재 유무 확인)은 안 쓰는거죠..?

이거랑 그 소상공인 데이터랑 헛갈려서요 ㅏ 소상공인 데이터는 업종 구분용 맞죠?

### ▼ 주변 가게 거리별 계산

#### 1. 전체 구조

[프론트엔드 (React 또는 JS)]

- └ navigator.geolocation → 사용자 GPS 가져오기
- └ Kakao Maps API → 지도 + 마커 + 내 위치 표시
- └ Spring API 호출 → /api/stores → 착한 가게 리스트 받아옴
- └ JS에서 거리 계산 → 가까운 5개 선별
- └ 마커 찍기 + 리스트 UI 표시

[백엔드 (Spring)]

- └ 가게 목록 조회 API → 위치정보 포함한 JSON 응답

#### 2. 프론트엔드(React) 코드

- HTML

```
<div id="map" style="width:100%; height:400px;"></div>
<ul id="store-list"></ul>
```

- JS (위치 + 지도 + 거리계산)

```
<script src="//dapi.kakao.com/v2/maps/sdk.js?appkey=YOUR_KAKAO_AP
<script>
navigator.geolocation.getCurrentPosition(function(pos) {
  const lat = pos.coords.latitude;
  const lon = pos.coords.longitude;

  const mapContainer = document.getElementById('map');
  const mapOption = {
```

```

    center: new kakao.maps.LatLng(lat, lon),
    level: 4
  };
  const map = new kakao.maps.Map(mapContainer, mapOption);

  // 내 위치 마커
  const userMarker = new kakao.maps.Marker({
    map: map,
    position: new kakao.maps.LatLng(lat, lon),
    title: "내 위치"
  });

  // 착한 가게 리스트 불러오기 (Spring API)
  fetch("/api/stores")
    .then(res => res.json())
    .then(stores => {
      const withDistance = stores.map(store => {
        const d = getDistance(lat, lon, store.lat, store.lng);
        return { ...store, distance: d };
      });

      // 가까운 순으로 정렬
      const closest = withDistance.sort((a, b) => a.distance - b.distance).slice(0, 5);

      // 마커 찍기
      closest.forEach(store => {
        new kakao.maps.Marker({
          map: map,
          position: new kakao.maps.LatLng(store.lat, store.lng),
          title: store.name
        });
      });

      // 리스트 표시
      const ul = document.getElementById('store-list');
      closest.forEach(store => {
        const li = document.createElement('li');
        li.innerText = `${store.name} (${store.distance.toFixed(2)} km)`;
      });
    });

```

```

        ul.appendChild(li);
    });
});
});

// Haversine 거리 계산 함수 (단위: km)
function getDistance(lat1, lon1, lat2, lon2) {
    const R = 6371; // 지구 반지름
    const dLat = (lat2 - lat1) * Math.PI / 180;
    const dLon = (lon2 - lon1) * Math.PI / 180;
    const a =
        Math.sin(dLat/2) * Math.sin(dLat/2) +
        Math.cos(lat1 * Math.PI / 180) * Math.cos(lat2 * Math.PI / 180) *
        Math.sin(dLon/2) * Math.sin(dLon/2);
    const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
    return R * c;
}
</script>

```

백엔드(Spring Boot) 코드

```

@GetMapping("/api/stores")
public List<StoreDTO> getStores() {
    return storeService.getAllStores(); // 각 store에 name, lat, lng 포함
}

```

#### ▼ 다이어그램들

0 블록다이어그램.drawio.pdf

1 소비인증 sd.drawio.pdf

2\_사용자제보sd.drawio.pdf

3 기부sd.drawio.pdf

4 플로우차트.drawio.pdf

5 시스템구조도.pdf

기부 시퀀스 업데이트.drawio.pdf

플로우차트 업데이트.drawio.pdf

ppt 초안 - 단락 텍스트 - 프레젠테이션

▼ 카테고리별 개수

|              |    |  |
|--------------|----|--|
| category     |    |  |
| 백반/한정식       | 27 |  |
| 카페           | 24 |  |
| 돼지고기 구이/찜    | 14 |  |
| 중국집          | 13 |  |
| 피자           | 13 |  |
| 경양식          | 13 |  |
| 미용실          | 11 |  |
| 치킨           | 11 |  |
| 국수/칼국수       | 10 |  |
| 빵/도넛         | 10 |  |
| 국/탕/찌개류      | 8  |  |
| 김밥/만두/분식     | 8  |  |
| 안경렌즈 소매업     | 6  |  |
| 일식 회/초밥      | 6  |  |
| 요리 주점        | 6  |  |
| 족발/보쌈        | 4  |  |
| 마라탕/튀김       | 4  |  |
| 입시·교과학원      | 4  |  |
| 해산물 구이/찜     | 3  |  |
| 일식 카레/돈가스/덮밥 | 3  |  |
| 핸드폰 소매업      | 3  |  |
| 횃집           | 3  |  |

|               |   |
|---------------|---|
| 해산물 구이/찜      | 3 |
| 횃집            | 3 |
| 슈퍼마켓          | 2 |
| 기타 의류 소매업     | 2 |
| 사진촬영업         | 2 |
| 미술학원          | 2 |
| 기타 기술/직업 훈련학원 | 2 |
| 기타 오락장        | 2 |
| 베트남식 전문       | 2 |
| 닭/오리고기 구이/찜   | 2 |
| 반찬/식료품 소매업    | 2 |
| 버거            | 2 |
| 음악학원          | 2 |
| 치과의원          | 2 |
| 레크리에이션 교육기관   | 2 |
| 일식 면 요리       | 1 |
| 내과/소아과 의원     | 1 |
| 피부 관리실        | 1 |
| 독서실/스터디 카페    | 1 |
| 외국어학원         | 1 |
| 정육점           | 1 |
| 소고기 구이/찜      | 1 |
| ...           |   |
| 요가/필라테스 학원    | 1 |
| 인테리어 디자인업     | 1 |
| 기타 한식 음식점     | 1 |
| 그 외 기타 종합 소매업 | 1 |

Name: count, dtype: int64

## 교수님 강조하신 부분 보고서에 담겨졌는가?

### ▼ 교수님 피드백

1. 신규성/진보성 요건 강화
  - o 영수증 OCR 인증을 통한 착한 소비 활동 기록
  - o 사용자 참여형 커뮤니티를 통한 착한 가게 제보 및 등록 시스템
  - o 포인트 적립 및 캐릭터 육성을 통한 기부 연계 메커니즘
  - o 위치 기반 가게 탐색 및 SNS 공유 기능을 통한 선한 영향력 확산
2. 시스템 구성을 아래 내용을 참고하여 구체화
  - o 사용자가 업로드한 영수증 이미지를 수신하고, OCR(optical character recognition) 기술을 통해 해당 이미지로부터 상호명, 거래 일자, 결제 금액 등의 텍스트 정보를 추출하는 소비 인증 모듈; Microsoft Azure OCR API를 사용하며, 이미지 내 기울기 보정, 대비 조정 등의 사전 이미지 전처리를 수행하고, 추출된 텍스트 데이터는 정규표현식 필터링을 통해 날짜, 금액, 가맹점명 항목으로 분리
  - o 위 추출된 상호명, 거래 일자 및 금액을 착한 가게 데이터베이스에 저장된 정보와 비교하여, 인증의 유효성을 판단하고, 인증 결과에 따라 정해진 포인트를 부여하는 소비 검증 및 포인트 부여 모듈
  - o 사용자가 착한 가게 정보를 포함하는 제보 게시글을 작성하고, 해당 게시글에 대해 다른 사용자들로부터 일정 수 이상의 '좋아요'와 일정 이하의 '싫어요' 비율을 획득했을 경우, 가게 등록 요청을 승인 대상으로 전환하고, 관리자의 최종 승인을 거쳐 착한 가게 데이터베이스에 등재하는 가게 제보 및 등록 모듈; 제보 게시글의 '좋아요 수'가 100 이상이며 '싫어요 비율'이 전체 반응 수의 50% 이하일 경우를 등록 심사 조건으로 하며, 게시글 작성 후 일정 시간(예: 72시간) 동안 집계된 커뮤니티 반응을 기준으로 관리자 검토 대상으로 분류
  - o 사용자의 인증 활동, 리뷰 활동, 투표 활동 등에 따라 누적된 포인트를 저장하며, 포인트가 사전에 정의된 누적값(예: 10,000점)에 도달하면, 사용자 계정에 연결된 가상 캐릭터의 성장 단계가 완료되며, 해당 포인트가 자동으로 사전에 지정된 기부처에 기부되도록 처리하는 기부 연계 포인트 관리 모듈; 포인트로 육성 가능한 가상 캐릭터를 단계별로 제공하며, 아기 → 청소년 → 성체 순으로 성장하며, 성장 완료 시점에 기부가 자동 수행되고, 다음 캐릭터 육성을 선택할 수 있는 기능을 포함
  - o 사용자의 실시간 GPS 좌표와 착한 가게의 위치 정보를 비교하여, 사용자 반경 2km 이내에 존재하는 착한 가게 리스트를 필터링하고, 이를 지도 기반 인터페이스로 시각화하여 사용자에게 제공하는 위치 기반 가게 탐색 모듈; 계절, 사회 이슈(예: 재난지역), 사용자 선호 업종 등을 고려하여 착한 가게 추천 리스트를 동적으로 구성하며, 해당 리스트를 지도 위에 업종별 마커 형태로 시각화하여 사용자가 직관적으로 탐색할 수 있도록 구성
- > 2 / 5 / 10 ~ 20 : (소도시) / (대도시) 몇 개를 띄울 건지 & 선정기준 -> 내 위치로부터 가까운 가게 3개만 선택해 노출하도록



(기본순 / 인기순/ ) 똑같은 가게 / 똑같은 순서 / 리스트(더보기칸 살리고)

- o 인증 완료된 소비 내역을 기반으로 HTML 구조의 인증 피드를 이미지로 변환 (html2canvas 활용)하고, 사용자에게 해당 이미지를 다운로드 가능한 형태로 제공함과 동시에, SNS 공유 메시지 및 해시태그를 자동 복사하도록 하여 사용자의 자발적 공유를 유도하는 소셜 공유 인터페이스 모듈
- o 사용자별로 소비 이력, 포인트 누적 현황, 기부 내역, 리뷰 및 댓글 활동, 제보 게시글 이력 등을 종합 관리하며, 기여도 순위 또는 선호도 기반 분석 결과를 표시하는 사용자 활동 이력 및 시각화 모듈

---

## 회의 결과

1. (시연) 블록다이어그램 교수님 피드백 반영 / 유스케이스 분리? / 기부SqD에 주스탠드 추가 / 3.5 시스템 아키텍처 업데이트 (노션 백엔드 탭 보고) / 시스템구조도 교수님 피드백 반영
2. 마이페이지 - 기부 내역 페이지만 살리기. 나머지는 디자인상만 표시

---

## 다음 회의

일정: 202Y년 MM월 DD일 00:00

준비: [다음 회의 때 준비해야 할 사항]