# R&B

빠라메

## 5조 발표

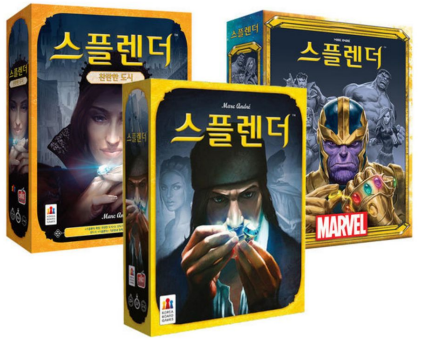
김우현, 김현홍, 전도명,
정원이, 조안나

# 목차

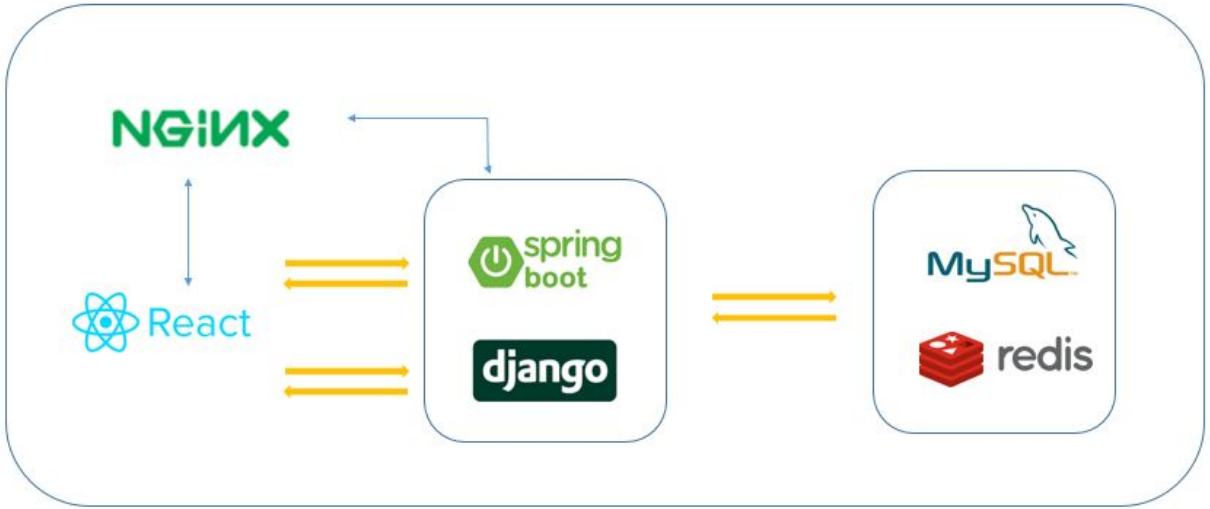
# R&B란?

Recommend Boardgames

사용자 취향에 알맞는 보드게임 추천 서비스




# 아키텍쳐



# 기능 명세서

| 서비스-메뉴 | 필요 기능(story) | 우선순위 | 기능 설명 | 상세 | 스토리포인트(FE) | 진행상황/BE | 진행상황/FE | 레퍼런스 |
|---|---|---|---|---|---|---|---|---|
| 회원가입 | 일반 회원가입 | 2 | 이메일아이디 / 비밀번호 / 비밀번호 확인 | 회원가입 하려면 이메일, 비밀번호, 닉네임, 프로필 사진을 입력해야한다.<br>비밀번호는 8~16자리 알파벳, 숫자의 조합으로 입력받는다. | | | | |
| | 세부정보 입력 | 2 | 사용자 닉네임, 이미지(미정) 입력 | 닉네임도 이메일과 같이 중복검사를 진행한다.<br>입력 내용이 변경될때마다 중복체크.<br>닉네임도 2~8글자 한글, 영문, 숫자 조합으로 입력받는다. | | | | |
| | 이메일 인증 | 2 | 이메일 아이디로 인증코드 발송, 인증코드 검증 후 회원가입 완료 | | | | | |
| | 가입 완료 페이지 | 2 | 회원가입 완료 확인 및 로그인 페이지로 이동 | | | | | |
| 로그인 | 일반 로그인 | 2 | 이메일아이디, 비밀번호로 로그인 | 로그인 실패 시 모달창으로 실패 메시지를 띄운다 | | | | |
| | SNS로그인 | 1 | 카카오, 네이버, 구글등 SNS아이디를 이용한 로그인 | 아이콘 버튼 | | | | |
| 로그아웃 | 로그아웃 | 1 | 비회원상태로 전환 | | | | | |
| 아이디/비밀번호 찾기 | 비밀번호 찾기 | 2 | 이메일아이디로 새 비밀번호 설정 링크 전송 | 링크를 통해 새로운 비밀번호를 설정하는 페이지로 이동 | | | | |
| 마이페이지 | 찜 목록 | 2 | 찜해놓은 보드게임 리스트 열람 | 찜해놓은 보드게임 무한스크롤로 출력 | | | | |
| | 가지고 있는 보드게임 | 2 | 내가 소유중인 보드게임 리스트 관리 | 소유중인 보드게임 무한스크롤로 출력 | | | | |
| | 친구관리 | 3 | 친구 리스트, 쪽지 보내기 | 양방향으로 어떻게 구현할지 | | | | |
| | 메시지함 | 3 | 받은 쪽지 확인 | | | | | |
| | 리뷰 관리 | 2 | 내가 등록한 리뷰 모아보기 | 각 리뷰 카드화 | | | | |
| | 최근에 열람한 보드게임 | 2 | 최근에 상세보기를 누른 보드게임 리스트 | 캐러셀 슬라이드로 최근 열람한 보드게임 10개 구현 | | | | |
| 추천 | 보드게임 취향 선택 | 1 | 초기 로그인시 추천 시스템 초기 설정 | 회원 테이블에 최초 로그인인지 확인할 수 있는 컬럼을 만들어 체크하고<br>최초 사용자이면 인기순으로 정렬한 보드게임 중 3개 이상 선택하는<br>페이지로 넘어간다 | | | | |
| | 비회원 - 인기순 정렬 | 1 | 골드스타트 상태의 비회원은 단순 인기순 정렬 | | | | | |
| | 회원 - 컨텐츠기반 필터링 | 1 | 초기 설정이 완료된 회원은 초기 설정과 로그를 통해<br>컨텐츠 기반 필터링을 통한 보드게임 추천 제공 | 사용자 행동 로그를 어떻게 쌓을 것인가(상세보기 클릭 이벤트?,<br>페이지에 머무르는 시간? 설명 유튜브 시청 시간? 등등) | | | | |
| 보드게임 검색 | 필터링 | 1 | 장르, 인원수, 소요시간, 난이도 | <추천이랑 연동 > 장르(체크박스), input range, range, range | | | | |
| | 정렬 | 1 | 평점순, 리뷰 개수 순으로 정렬 | 리스트 | | | | |
| 상세보기 | 유튜브연동 | 1 | 게임 소개, 플레이 방법 유튜브 링크 | 보드게임 플레이 방법 유튜브 영상 임베드 | | | | https://developers.google.com/youtube/v3/docs/search?hl=ko |
| | 네이버 쇼핑 연동 | 1 | 네이버 쇼핑 검색 API로 게임 구매 시 검색 | 우리 사이트에서 검색 결과 목록 보여주기 or 그냥 링크만 띄워주기 | | | | https://developers.naver.com/docs/search/shopping/ |
| | 기본정보 | 1 | 사진, 이름, 게임 소개, 플레이 인원수, 소요시간, 장르 | | | | | |
| | 찜하기 | 1 | 찜목록에 추가(버튼형식) | 사라지는 조그마한 alert창으로 추가, 삭제 안내 | | | | |
| | 갖고있어요 | 1 | 마이페이지에 가지고있는 보드게임에 추가(버튼형식) | 사라지는 조그마한 alert창으로 추가, 삭제 안내 | | | | |
| | 별점 | 1 | 드래그 형식으로 간편하게 | | | | | |
| | 난이도 | 1 | 투표형식 도입(일정 등급이상) | | | | | |
| | 비슷한 장르의 게임 | 1 | 비슷한 스타일 (인원수, 소요시간, 장르 등) | 이게임 한사람은 이런게임도 했다 캐러셀 슬라이드로 추천 | | | | |
| 후기 페이지 | 사용자 제품 후기 | 2 | 플레이해본 보드게임 후기 작성, 열람 | | | | | |
| 중고거래 | 판매,구매 | 3 | 보드게임 판매, 구매 게시물 등록 | 카테고리를 통해 판매 구매 선택, 상품 이미지 등록, 가격 상세정보 등록,<br>거래방법 선택 / 거래완료 설정 | | | | |
| | 나눔 | 3 | 보드게임 나눔 게시물 등록 | 게임 이미지 등록, 상세정보 등록, 전달방법 선택 / 나눔 완료 선택 | | | | |
| | 교환 | 3 | 게임 등록로 교환 or 게임 부품 잃어버린거 교환 | | | | | |
| 같이할사람 구하기 | 구인 | 3 | 같이 플레이할 사람 구하기 | 지도 영역 클릭하면 자동으로 지역 필터링 | | | | |
| | 보드게임카페 지도 | 3 | 토글 버튼으로 주위 카페와 구인하는 지역 표시 | 지도상에 마커로 표시된 보드게임카페 클릭하면 상세보기 모달 출력 | | | | |
| 보드게임 카페 게시판 | 갔다온 후기(사용자) | 3 | 본인이 다녀온 보드게임 카페 후기 작성 | 보드게임카페 상세보기 모달 아래에 리스트형식으로 출력 | | | | |
| | 사장님 | 3 | 카페 홍보 목적 | 게시판 형식 | | | | |
| 문의하기 | 사이트에 없는 게임 등록 문의 | 4 | 게임 등록 요청(상세보기 페이지에서 제공하는 정보 모두 포함)<br>=> 관리자가 검토 후 등록 | 챗봇 | | | | |
| 관리자페이지 | 게임 등록 | 5 | 게임 등록 요청을 검토하여 게임 등록 | | | | | |
| | 회원관리 | 5 | 회원정보 수정, 탈퇴 등 기본 기능 + 통계 | | | | | |

zz.ipynb > df (19230, 9)

| | index | id | image | gamena… | yearpu… | minpla… | maxpla… | minpla… | maxpla… | boardgamecategory |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 30549 | https://cf.geekdo-i… | Pandemic | 2008 | 2 | 4 | 45 | 45 | Medical |
| 1 | 1 | 822 | https://cf.geekdo-i… | Carcasso… | 2000 | 2 | 5 | 30 | 45 | City Building\|Medieval\|Terri… |
| 2 | 2 | 13 | https://cf.geekdo-i… | Catan | 1995 | 3 | 4 | 60 | 120 | Economic\|Negotiation |
| 3 | 3 | 68448 | https://cf.geekdo-i… | 7 Wonders | 2010 | 2 | 7 | 30 | 30 | Ancient\|Card Game\|City Build… |
| 4 | 4 | 36218 | https://cf.geekdo-i… | Dominion | 2008 | 2 | 4 | 30 | 30 | Card Game\|Medieval |
| 5 | 5 | 9209 | https://cf.geekdo-i… | Ticket t… | 2004 | 2 | 5 | 30 | 60 | Trains |
| 6 | 6 | 178900 | https://cf.geekdo-i… | Codenames | 2015 | 2 | 8 | 15 | 15 | Card Game\|Deduction\|Party Ga… |
| 7 | 7 | 31260 | https://cf.geekdo-i… | Agricola | 2007 | 1 | 5 | 30 | 150 | Animals\|Economic\|Farming |
| 8 | 8 | 3076 | https://cf.geekdo-i… | Puerto R… | 2002 | 3 | 5 | 90 | 150 | City Building\|Economic\|Farmi… |
| 9 | 9 | 40692 | https://cf.geekdo-i… | Small Wo… | 2009 | 2 | 5 | 40 | 80 | Fantasy\|Fighting\|Territory B… |
| 10 | 10 | 167791 | https://cf.geekdo-i… | Terrafor… | 2016 | 1 | 5 | 120 | 120 | Economic\|Environmental\|Indus… |
| 11 | 11 | 70323 | https://cf.geekdo-i… | King of … | 2011 | 2 | 6 | 30 | 30 | Dice\|Fighting\|Movies / TV / … |
| 12 | 12 | 14996 | https://cf.geekdo-i… | Ticket t… | 2005 | 2 | 5 | 30 | 60 | Trains |
| 13 | 13 | 148228 | https://cf.geekdo-i… | Splendor | 2014 | 2 | 4 | 30 | 30 | Card Game\|Economic\|Renaissan… |
| 14 | 14 | 2651 | https://cf.geekdo-i… | Power Gr… | 2004 | 2 | 6 | 120 | 120 | Economic\|Industry / Manufact… |
| 15 | 15 | 173346 | https://cf.geekdo-i… | 7 Wonder… | 2015 | 2 | 2 | 30 | 30 | Ancient\|Card Game\|City Build… |
| 16 | 16 | 129622 | https://cf.geekdo-i… | Love Let… | 2012 | 2 | 4 | 20 | 20 | Card Game\|Deduction\|Renaissa… |
| 17 | 17 | 169786 | https://cf.geekdo-i… | Scythe | 2016 | 1 | 5 | 90 | 115 | Economic\|Fighting\|Science Fi… |

## 마이그레이션

- csv 파일을 불러와 필요한 부분만 정제하여 pickle로 객체 저장
- 파싱된 데이터는 분석과정에서 데이터프레임화

hp.ipynb > review (15823269, 3)

| | index | user | rating | board |
|---|---|---|---|---|
| 0 | 0 | Xorph | 8 | 176 |
| 1 | 1 | catmogwai | 7.25 | 111417 |
| 2 | 2 | Pura_Vida | 5.5 | 205418 |
| 3 | 3 | Geeken | 3 | 272438 |
| 4 | 4 | vobacbe | 7 | 205637 |
| 5 | 5 | zollom04 | 5 | 478 |
| 6 | 6 | Dr Jacka… | 6 | 63539 |
| 7 | 7 | airxmoa | 7 | 2719 |

# SUB2 진행

## 컨텐츠기반 필터링

- [x] 장르를 벡터화 한 뒤 이를 기반으로 코사인 유사도 측정
구해진 유사도 행렬을 정렬하여 아이템마다 연관된 장르의 게임 순서대로 정렬

```
def content_based_recommend(dataframes):
    """
    컨텐츠 기반 필터링
    """
    count_vect = CountVectorizer(min_df=0, ngram_range=(1,2))
    genre_match = count_vect.fit_transform(dataframes['boardgamecategory'])
    # 장르 문자열 벡터화
    genre_sim = cosine_similarity(genre_match, genre_match)
    """ 벡터화된 정보를 기준으로 코사인 유사도 측정
    genre_sim = 유사도 행렬 genre_sim[i][j] = i번째 아이템과 j번째 아이템이 얼마나 비슷한지(0~1)
    [[1.         0.         0.         ... 0.         0.         0.        ]
    [0.         1.         0.         ... 0.         0.53935989 0.        ]
    [0.         0.         1.         ... 0.         0.         0.        ]
    [0.         0.3344968  0.16012815 ... 0.         0.12403473 0.07161149]
    [0.         0.13483997 0.         ... 0.         0.         0.11547005]]
    """
    genre_sim_sorted_ind = genre_sim.argsort()[:, ::-1]
    print(genre_sim_sorted_ind[:4])
    """
    genre_sim_sorted_ind = 비슷한 순으로 아이템 인덱스 출력
    [[    0 18638 18455 ... 12807 12808  9614] 0번째 아이템과 가장 비슷한 순으로 출력 =dataframes.iloc[18638]> dataframes.iloc[18455] ...
    [ 8372  2344  7481 ... 12637 12638     0] 1번째 아이템과 가장 비슷한 순으로 출력 =dataframes.iloc[8372]> dataframes.iloc[2344] ...
    [ 8386  7182 16589 ... 12630 12631     0]   ...
    [    3    15 12751 ... 10734 10732     0]]
    """
```

```python
def collaboration_filtering(dataframes):
    # review = pd.read_pickle("../data/review.pkl")
    review = dataframes['reviews']
    review = review.astype({'rating': 'float'}) # 타입 변환

    review = review[['user', 'rating', 'board']] # 필요한 데이터만 정제

    review['Count'] = review.groupby(['user'])['rating'].transform('count') #리뷰가 적은 유저
    game_user_rating = review.pivot_table('rating', index = 'board', columns='user') # 보
    review_dic=game_user_rating.to_dict() #추천시스템 적용을 위해 딕셔너리로 변환

    #NaN값 제거
    review_dic_drop=dropna(review_dic)
    print(top_match(review_dic,"1000daffyducks",5)) #입력 유저와 연관성이 높은 유저 5명 출력

# NaN값 제거
def dropna(data): # name : i , movie : j
    for i in data:
        for j in data[i]:
            name=i
            movie=j
            value=data[i][j]
        data[i]={movie: value for movie, value in data[i].items() if pd.isnull(value)==False}
```

**협업 필터링**

- 유저 X 게임 데이터프레임을 딕셔너리화
  => 피어슨 상관계수를 통해 KNN알고리즘으로 성향이 비슷한 유저 데이터 확보

```
def cos_sim(X,Y):
    return dot(X,Y)/((norm(X)*norm(Y))+1e-7)

def top_match_ar2(data, name, rank=5,simf=cos_sim):
    sim=[]
    for i in range(len(data)):
        if name != i:
            sim.append((simf(data[i],data[name]),i))
    sim.sort()
    sim.reverse()
    return sim[:rank]
```

```
data["boards"]["boardgamecategory"].isnull().sum() #135
data["boards"]["boardgamecategory"]=data["boards"]["boardgamecategory"].fillna('')
data["boards"]["boardgamecategory"].isnull().sum() #0 으로 바뀜 내적하면 모두 0 나옴

tfidf=TfidfVectorizer(stop_words='english')
tfidf_mat=tfidf.fit_transform(data["boards"]["boardgamecategory"]).toarray()

boardList = []
for sim, board_id in top_match_ar2(tfidf_mat,0,10):
    boardList.append((sim, data["boards"].loc[board_id,'gamename']))
print(boardList[:10])
```

```
[(0.9999999000000099, 'Medicine Whoops'), (0.9999999000000099, 'Infection'), (0.9999999000000099, 'Code Triage'), (0.9999999000000099, 'G.Nom
(0.9999999000000099, 'Medical Frontier'), (0.9999999000000099, 'Intern'), (0.9999999000000099, 'Pandemic: Hot Zone - North America'), (0.9999
0000099, 'Quarantine'), (0.9999999000000099, 'Pandemic: Iberia'), (0.9128747038267573, 'Treatment: A Psychiatry Card Game')]
```

## 장르별 추천시스템

- tf-idf행렬을 생성하여 단어의 빈도수를 나타내고 배열로 변경후 코사인 유사도를 구하는 함수를 통해 해당 보드게임과 가장 유사한 순서로 추천해주는 시스템입니다.

# 향후 계획

1. API 명세서 꼼꼼히 작성!

   중간에 API 수정 최소화

2. 배포 먼저!

   한명의 팀원을 더 얻는 효과

3. 캐싱, 보안, 테스트

   공통때 못해본 기술 적용해보기!

4. 추천 알고리즘 계속 다듬기!

   무한 분석

이상으로 발표를 마칩니다

질문 받아요~!

★★★★★

감사합니다!

THANK YOU