



골라먹자

음식 랜덤 선택 앱

선정 배경

자취생



주부

직장인

하루 세끼 메뉴좀 쉽게 정하자!!!!

골라
먹자

랜덤 골라

먹고 싶은 거 없을 때
세상 모든 음식 중에 골라!

내 리스트에서
골라

먹고 싶은거 많을 때 골라!

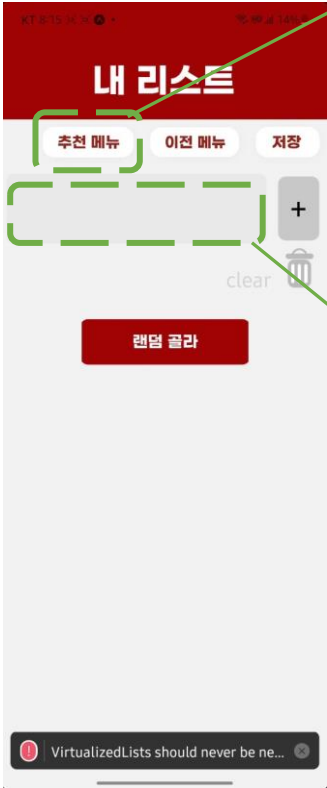
앱 사용설명 #1



앱 사용설명 #2



먹고 싶은 메뉴가 많을 때
<내 리스트에서 골라> 선택

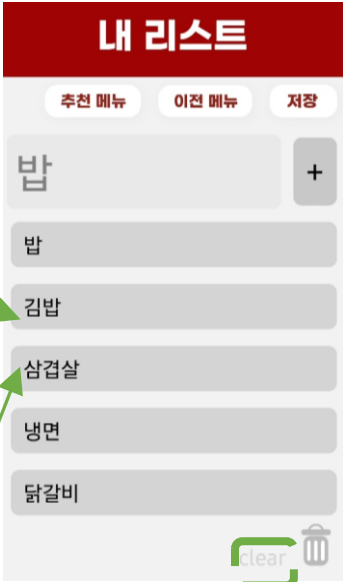


내가 원하는 메뉴를 나열할 수 있음
1) 추천메뉴에서 삽입
2) 직접 입력

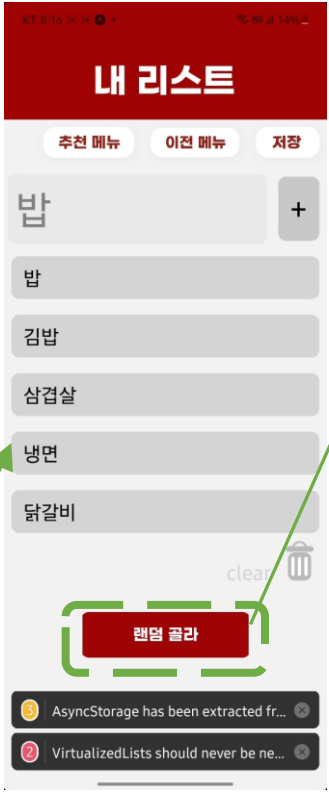
추천메뉴에서
원하는 메뉴 클릭,
내 메뉴에 삽입



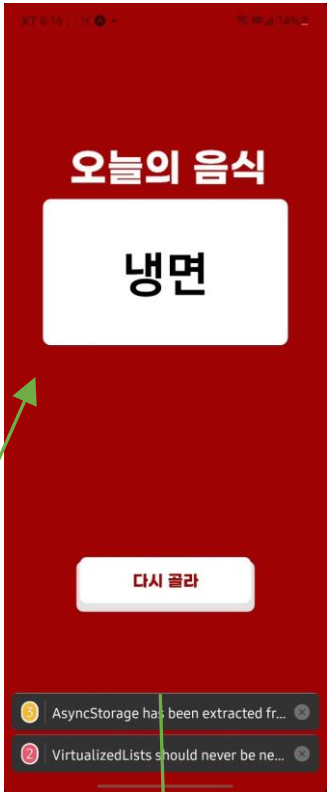
직접 입력하여
내 메뉴에 삽입



Clear를 눌러
내 메뉴 모두 초기화
Or
하나씩 휴지통을 눌러
삭제



이전메뉴를 선택해
전에 저장한 메뉴
다시 불러옴



데모영상

(따로 파일로도 제출했습니다)



보완한 계획(발표했던 계획)

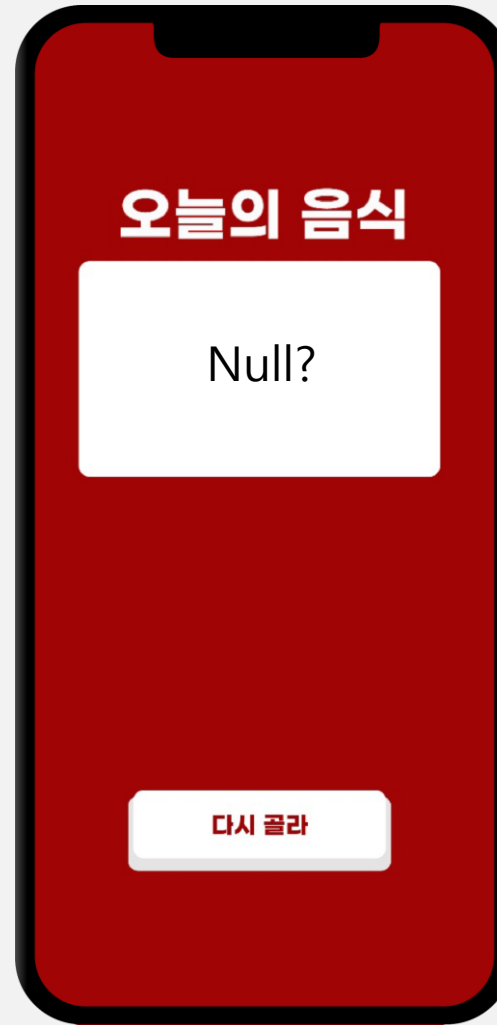
1



< 다시 골라> 버튼을 누를 때,
한번 나왔던 메뉴는
중복해서 나오지 않도록!

-> 수정했습니다!
(12월 18일 기준)

2

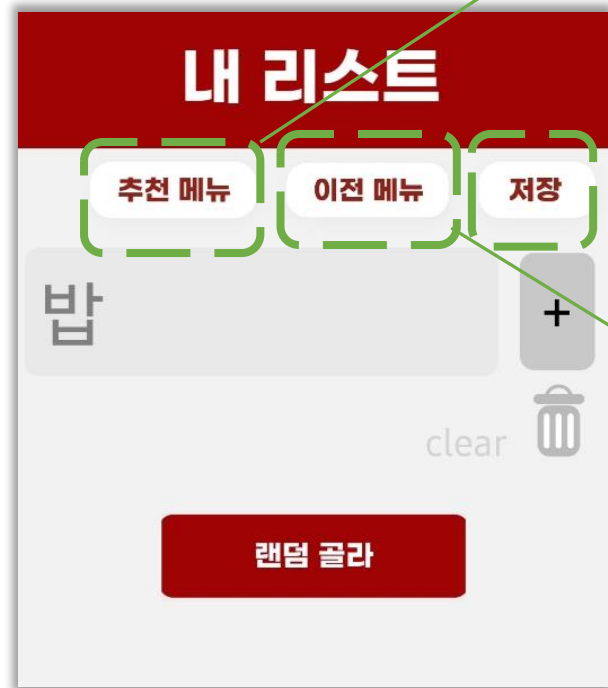


user_menu에 남은 원소가
없을 때,
예외처리!

-> 수정했습니다!
(12월 18일 기준)

예외처리는 자꾸 오류가 나서
경고창을 띄우는 방식으로 대처했습니다.

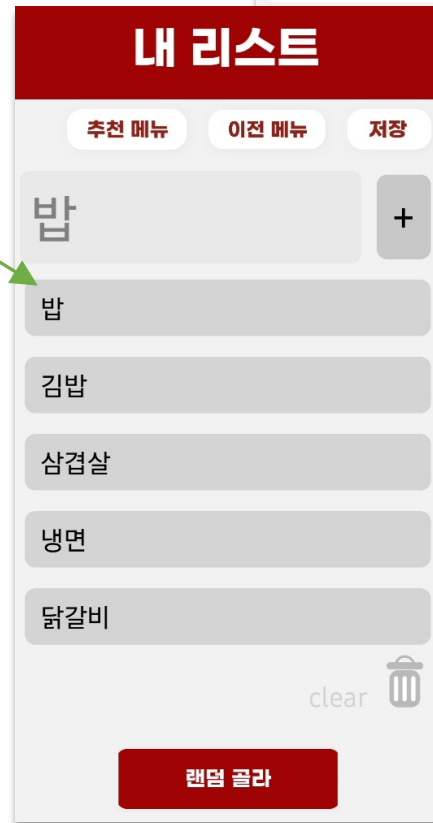
추가한 사항 (발표 이후에 추가)



추천 메뉴

+ 김치찌개
+ 김밥
+ 삼겹살
+ 냉면
+ 닭갈비
+ 매운탕

추천메뉴를 통해
db에 저장되어 있는
메뉴들을 보여주고
해당 메뉴를 클릭하여
쉽게 내 리스트에
추가할 수 있게 함!



저장, 이전메뉴 를 통해 예
전에 작성했던 내 리스트를
저장하거나 불러올 수 있음!

특징점 #1 - 필터링 알고리즘

서버를 대체한 const 변수

```
const menu=[
  {name:'김치찌개',카테고리:'한식',알려지:'',특징:['매움','뜨거움'],육류:'소'},
  {name:'김밥',카테고리:'한식',알려지:'',특징:[],육류:''},
  {name:'삼겹살',카테고리:'한식',알려지:'',특징:['뜨거움'],육류:'돼지'},
  {name:'냉면',카테고리:'한식',알려지:'',특징:['차가움'],육류:''},
  {name:'닭갈비',카테고리:'한식',알려지:'',특징:['매움','뜨거움'],육류:'닭'},
  {name:'매운탕',카테고리:'한식',알려지:'해산물',특징:['뜨거움','매움'],육류:'생선'},
  {name:'비빔밥',카테고리:'한식',알려지:'계란',특징:['매움'],육류:''},
  {name:'냉라면',카테고리:'한식',알려지:'',특징:['매움','차가움'],육류:''},
  {name:'국밥',카테고리:'한식',알려지:'',특징:['뜨거움'],육류:'소'},
  {name:'반미',카테고리:'베트남',알려지:'',특징:['패스트푸드'],육류:''},

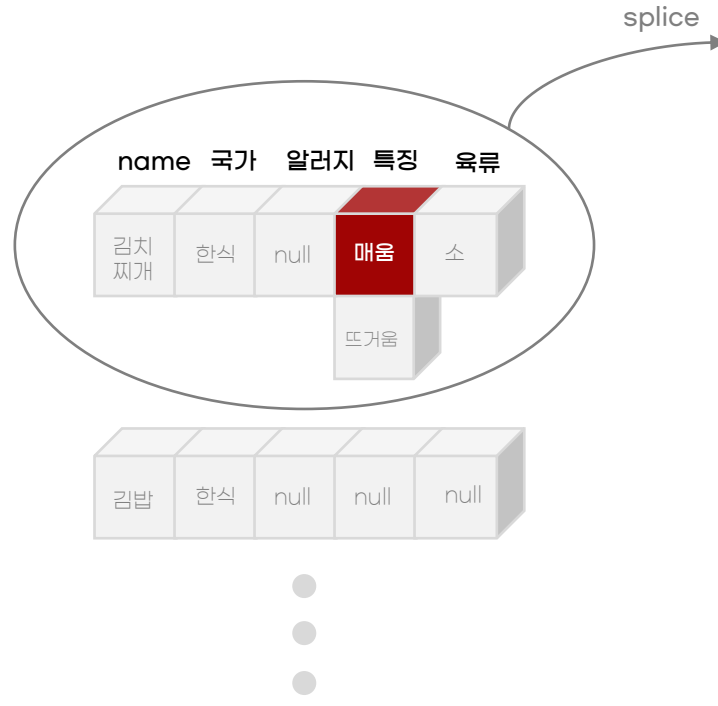
  {name:'커리',카테고리:'인도',알려지:'',특징:['뜨거움'],육류:''},
  {name:'탄두리치킨',카테고리:'인도',알려지:'',특징:['뜨거움'],육류:'닭'},
  {name:'인도식볶음밥',카테고리:'인도',알려지:'견과류',특징:['뜨거움'],육류:'닭'},
```



const변수로 40여개의 음식에 대해 menu라는 배열로 정리

user menu라는 새로운 배열에 menu배열을 복사하고, 그 중 기피되는 음식카테고리가 있는 배열의 원소는 splice 메소드로 삭제

```
var user_menu=[];
```

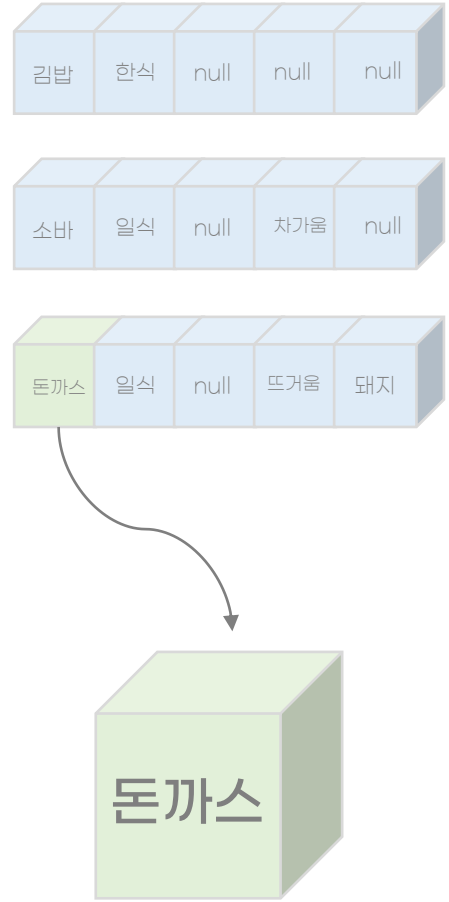


```
for(j=0;j<10;j++){
  for(i=0;i<user_menu.length;i++){
    if(user_menu[i].카테고리==props.thing || user_menu[i].알려지==props.thing || user_menu[i].육류==props.thing){
      user_menu.splice(i,1);
    }
  }
}
for(j=0;j<3;j++){
  for(i=0;i<user_menu.length;i++){
    for(k=0;k<user_menu[i].특징.length;k++){
      if(user_menu[i].특징[k]==props.thing){
        user_menu.splice(i,1);
      }
    }
    else if(user_menu[i].특징[k]==null){
      continue;
    }
  }
}
```

난수를 생성하여 남은 user menu 중 하나를 추천

```
var num=Math.floor(Math.random()*(user_menu.length));
```

```
<Text style={{fontSize:50,fontWeight:'bold'}}>
  {user_menu[num].name}</Text>
```



특징점 #2

1 효율적인 코드

전체적인 코드에서 배열은 `const menu` 배열과 `user_menu`만 사용하였습니다. 메인 화면에서 <랜덤골라> 나 <내 리스트에서 골라> 모두에서 `user_menu`만 사용하여 불필요한 메모리 사용을 줄였습니다.

또한 이 효율적인 배열 사용을 통해 <랜덤골라>나 <내 리스트에서 골라> 상관없이 마지막 화면 (난수를 생성해 해당 인덱스를 출력하는 화면)를 중복코드 없이 사용할 수 있었습니다.

2 직관적인 디자인

코드로 의도했던 깔끔한 디자인이 나오지 않아, 직접 일러스트로 로고, 배경등을 디자인해 해당 이미지를 앱에 올려 사용하였습니다.

덕분에 직관적인 메뉴들과 한눈에 들어오는 색감을 표현할 수 있었습니다.



감사합니다

한 학기동안 수업해주시고,
앱 제작에 많은 도움을 주신
김명준 교수님과 김민수 교수님께
감사합니다!