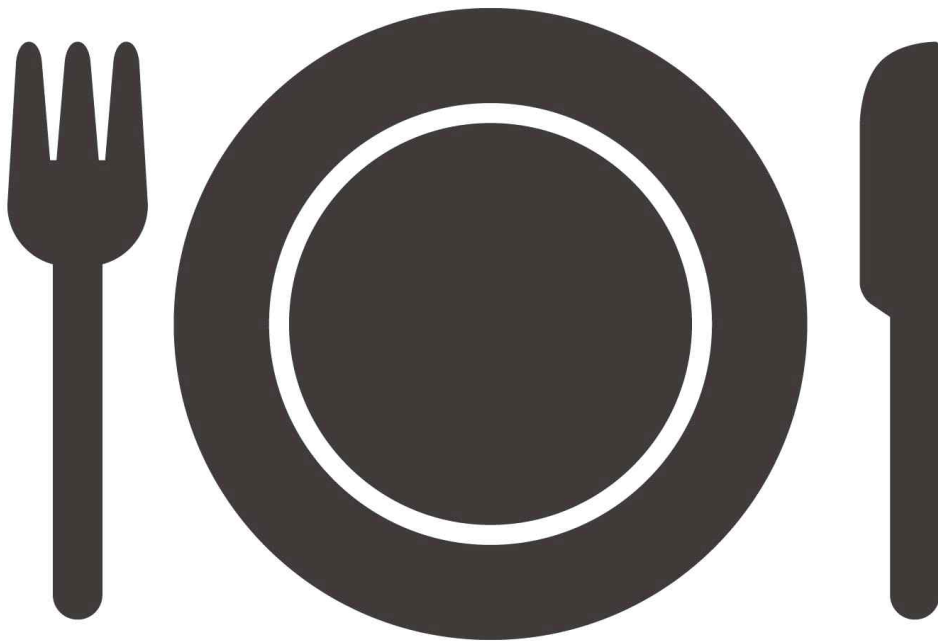


# 전 세계의 음식들

-Conceptualization document-



# [ Revision history ]

Revision date	Version #	Description	Author
	#1.1	Use case, concept of operation 수정	
	#1.2	System context diagram, use case, concept of operation 수정	

= Contents =

1. Business purpose .....	1
2. System context diagram .....	2
3. Use case list .....	3
4. Concept of operation .....	6
5. Problem statement .....	11
6. Glossary .....	12
7. References .....	13

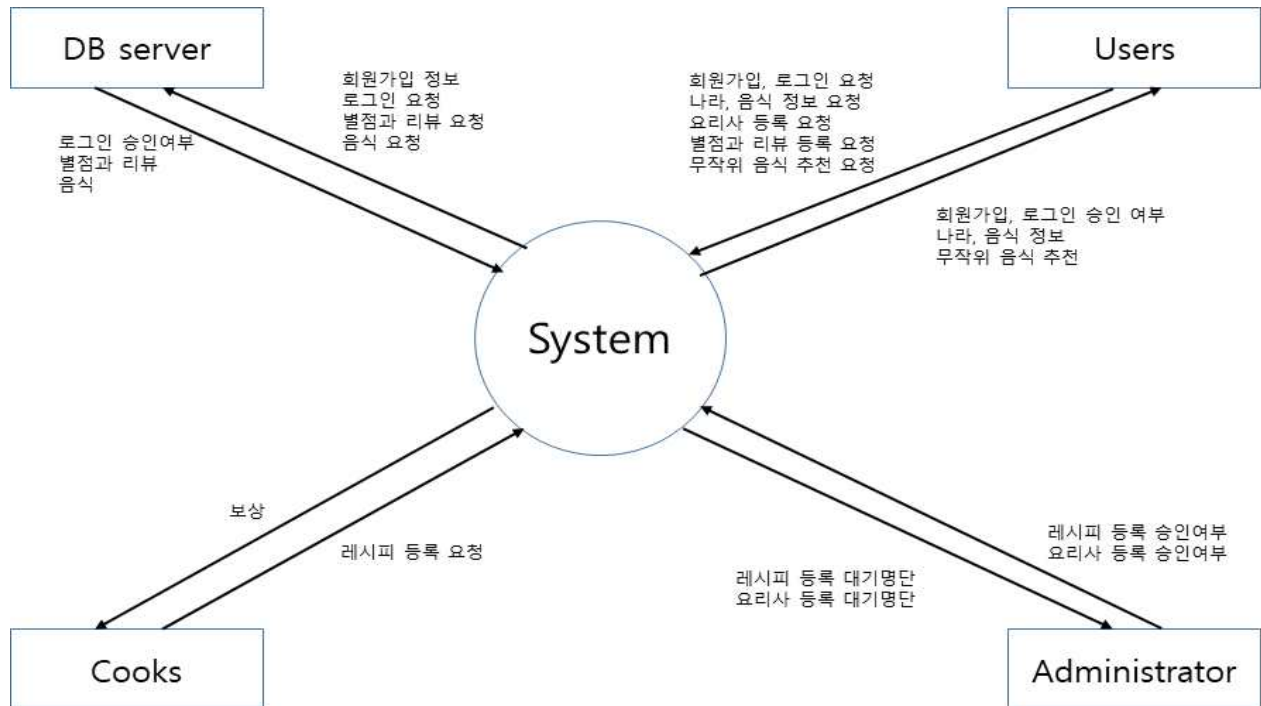
## 1. Business purpose

세상에는 요리 조리법을 알려주는 많은 앱이 있다. 하지만 대부분의 앱들은 단지 어떻게 음식을 만드는지를 알려주기만 하는 단조로운 패턴을 가지고 있다. 이런 앱과 차별화를 가진 조리법 앱을 만들면 좋겠다는 생각을 했고 그 결과로 ‘전 세계의 음식들’ 프로젝트를 구상했다. ‘전 세계의 음식들’은 나라별 이색 조리법을 알려주는 앱이다. 앞서 언급했듯이 대부분의 요리 앱의 메인 화면은 조리법을 한식, 양식과 같은 음식의 종류별로 나열한다. 지역별로 정렬한 소수의 앱이 있긴 하지만 이 역시 음식으로 유명한 나라만 다룬다거나 대륙별로 정렬을 해 특정 나라만의 특색을 알 수 없는 경우가 대부분이다. 이에 반해 ‘전 세계의 음식들’은 음식을 나라별로 정렬하고 음식으로 유명하지 않더라도 그 나라만의 고유 음식의 조리법을 제공하여 각 나라의 특색을 알 수 있게 한다.

코로나 19로 인해 더 많은 사람들이 집에서 요리를 하고 있다.<sup>1)</sup> 이를 통해 특히 지금과 같이 코로나 시국으로 여행을 가는 것이 제한됐을 때 집에서 전 세계의 원하는 나라의 고유 음식들을 직접 만들어보고 먹어보는 경험을 하는 것은 사용자로 하여금 신선함을 불러일으킬 것이다. 이것이 ‘전 세계의 음식들’의 가장 큰 목적이다. 오픈 API를 이용해 음식의 재료들의 가격대와 정보를 쉽게 확인하게 함으로써 편리성을 제공하는 것을 또한 목표로 한다. 요리를 좋아하는 누구든지 이 앱의 타겟층이 될 수 있다.

## 2. System context diagram

이 프로젝트를 위해 작성한 system context diagram은 아래와 같다.



[그림 2-1] System context diagram

음식, 별점과 리뷰, 회원정보를 저장하기 위한 DB 서버로 웹호스팅을 이용한다. 그 외의 용어들은 일상 생활에서 자주 쓰이는 단어들이라 직관적으로 이해할 수 있을 것이다. 만약 그렇지 않다면 6. Glossary를 참고하길 바란다.

### 3. Use case list

이 프로젝트에서 발견할 수 있는 use cases는 다음과 같다.

#### 1) 회원가입

Actor	Users, Cooks
Description	유저는 시스템의 특정 기능을 사용하기 위해 회원 등록한다.

#### 2) 로그인

Actor	Users, Cooks
Description	각자 자신의 아이디로 로그인한다. 시스템은 DB 서버한테 로그인 정보를 주고 회원정보를 받아 로그인 성공 여부를 반환한다.

#### 3) 로그아웃

Actor	Users, Cooks, An administrator
Description	로그아웃한다.

#### 4) 나라 조회

Actor	Users, Cooks
Description	간략한 설명이 함께 있는 나라의 리스트를 본다.

#### 5) 음식 리스트 조회

Actor	Users, Cooks
Description	나라 리스트에서 그 나라의 음식 리스트를 본다.

#### 6) 음식 조회

Actor	Users, Cooks
Description	나라 리스트에서 원하는 음식 선택 후 음식의 정보를 본다.

#### 7) 음식 조리법 조회

Actor	Users, Cooks
Description	음식 조리법을 본다.

#### 8) 요리사 등록 요청

Actor	Users, Cooks
Description	사용자가 요리사가 되기 위해 요리사 등록을 요청한다. 관리자가 허가한다면 해당 사용자는 요리사가 되며 조리법을 등록할 수 있다.

#### 9) 음식 등록 요청

Actor	Cooks
Description	요리사가 음식 등록을 요청한다. 관리자가 허가한다면 음식이 등록된다.

#### 10) 조리법 등록 요청

Actor	Cooks
Description	요리사가 조리법을 등록을 요청한다. 관리자가 허가한다면 조리법이 등록된다.

#### 11) 음식 추천

Actor	Users, Cooks
Description	취향별/무작위 음식 추천을 한다.

#### 12) 요리사 등록 관리

Actor	An administrator
Description	요리사 등록 요청을 승인할지 말지 결정한다.

#### 13) 음식 등록 관리

Actor	An administrator
Description	음식 등록 요청을 승인할지 말지 결정한다.

#### 14) 리뷰 조회

Actor	Users, Cooks
Description	리뷰를 조회한다.

#### 15) 리뷰 추가

Actor	Users, Cooks
Description	리뷰를 추가한다.



## 4. Concept of operation

이 장은 use cases를 어떻게 작동할지에 대한 내용을 기술한 장이다. 자세한 작동 개념은 다음과 같다.

### 1. 회원가입

<b>Purpose</b>	별점/리뷰 등록 및 조리법 등록 시에 식별 가능한 사용자 아이디가 있어야 한다.
<b>Approach</b>	아이디가 없으면 회원가입을 할 수 있게 한다.
<b>Dynamics</b>	앱을 처음 실행했을 경우와 별점/리뷰 등록 및 조리법 등록을 할 경우
<b>Goals</b>	회원가입 기능을 구현한다.

### 2. 로그인

<b>Purpose</b>	별점/리뷰 등록 및 조리법 등록 시에 식별 가능한 사용자 아이디가 있어야 한다.
<b>Approach</b>	앱을 실행하면 로그인을 할 수 있게 한다.
<b>Dynamics</b>	앱을 처음 실행했을 경우와 별점/리뷰 등록 및 조리법 등록을 할 경우
<b>Goals</b>	로그인 기능을 구현한다.

### 3. 로그아웃

<b>Purpose</b>	다른 아이디로 로그인하고 싶은 경우 먼저 로그아웃을 할 수 있어야 한다.
<b>Approach</b>	로그아웃할 수 있게 한다.
<b>Dynamics</b>	로그아웃을 하고싶은 경우
<b>Goals</b>	로그아웃 기능을 구현한다

### 4. 나라 조회

<b>Purpose</b>	사용자가 흥미를 끄는 나라를 고를 수 있도록 나라를 보여줘야 한다.
<b>Approach</b>	나라를 card view로 이름과 같이 가장 특색 있는 음식이나 그 나라의 특징을 보여준다.
<b>Dynamics</b>	나라별 특색 있는 조리법을 알고 싶은 경우
<b>Goals</b>	지도나 card view를 이용해 나라를 나열한다.

## 5. 음식 리스트 조회

<b>Purpose</b>	그 나라의 음식들의 리스트를 보여줘야 한다.
<b>Approach</b>	그 나라의 음식 이름들의 리스트를 보여준다.
<b>Dynamics</b>	나라별 특색 있는 조리법을 알고 싶은 경우
<b>Goals</b>	음식을 나열한다.

## 6. 음식 조회

<b>Purpose</b>	사용자의 요리 준비를 위해 필요한 재료와 가격대를 알려주면 사용자 입장에서 편안함을 느낄 것이다.
<b>Approach</b>	나라를 선택하면 그 나라의 음식들을 보여준다. 음식을 선택하면 재료를 보여준다.
<b>Dynamics</b>	음식의 재료와 가격대를 확인하고 싶은 경우
<b>Goals</b>	음식을 터치했을 때 재료가 나오고 재료를 터치했을 때 상세정보와 가격대를 표시한다.

## 7. 음식 조리법 조회

<b>Purpose</b>	만들고자 하는 음식의 조리법을 제공해야 한다.
<b>Approach</b>	한 단계씩 화면에 띄워 조리법을 보여준다.
<b>Dynamics</b>	음식의 조리법을 확인하고 싶은 경우
<b>Goals</b>	조리법을 단계별로 알려주는 화면을 만든다.

## 8. 요리사 등록 요청

<b>Purpose</b>	경력이나 자격 있는 유저에게 요리사로 등록될 자격을 줘서 조리법을 등록할 수 있게 하면 좋을 것 같다.
<b>Approach</b>	요리사 등록을 할 수 있는 탭을 만든다.
<b>Dynamics</b>	요리사 등록을 원할 경우
<b>Goals</b>	사용자가 검증 가능한 경력과 자격을 기입한 후 요리사 등록 요청을 할 수 있는 기능을 구현한다.

## 9. 음식 등록 요청

<b>Purpose</b>	운영자 외에 경력이 많거나 자격이 있는 유저가 음식을 올리면 종류가 풍부해질 것이다.
<b>Approach</b>	요리사에 한하여 음식등록 요청 탭을 추가한다.
<b>Dynamics</b>	요리사가 음식을 등록하고 싶을 경우
<b>Goals</b>	등록된 요리사들이 음식을 등록하는 기능을 구현한다.

#### 10. 조리법 등록 요청

<b>Purpose</b>	음식을 올리고 조리법도 같이 올려야 할 것이다.
<b>Approach</b>	요리사에 한하여 조리법 등록 요청 탭을 추가한다.
<b>Dynamics</b>	요리사가 조리법을 등록하고 싶을 경우
<b>Goals</b>	등록된 요리사들이 조리법을 등록하는 기능을 구현한다.

#### 11. 음식 추천

<b>Purpose</b>	음식을 무작위로 추천해주면 좋을 것이다.
<b>Approach</b>	음식을 무작위로 추천해주는 화면을 만든다.
<b>Dynamics</b>	무작위로 음식들을 보고 싶을 때
<b>Goals</b>	음식 추천을 보여주는 화면을 만든다.

#### 12. 요리사 등록 관리

<b>Purpose</b>	요리사 등록 요청을 시스템이 처리하면 애로사항이 클 것이다. 관리자가 이를 관리하면 좋을 것 같다.
<b>Approach</b>	관리자에게 등록 대기 탭을 만들어 승인/거부 결정을 내린다.
<b>Dynamics</b>	요리사 등록을 결정할 경우
<b>Goals</b>	관리자가 등록 여부를 직접 관리함으로써 신뢰성과 일관성 있는 처리를 한다.

#### 13. 음식 등록 관리

<b>Purpose</b>	음식 등록 요청을 시스템이 처리하면 애로사항이 클 것이다. 관리자가 이를 관리하면 좋을 것 같다.
<b>Approach</b>	관리자에게 등록 대기 탭을 만들어 승인/거부 결정을 내린다.
<b>Dynamics</b>	조리법 등록을 결정할 경우
<b>Goals</b>	관리자가 등록 여부를 직접 관리함으로써 신뢰성과 일관성 있는 처리를 한다.

#### 14. 리뷰 조회

<b>Purpose</b>	다른 사람이 적은 리뷰를 볼 수 있으면 좋을 것 같다.
<b>Approach</b>	다른 사람이 등록한 별점과 리뷰를 가져와서 보여준다.
<b>Dynamics</b>	다른 사람의 후기를 보고 싶을 경우
<b>Goals</b>	리뷰를 볼 수 있는 화면을 만든다.

#### 15. 리뷰 추가

<b>Purpose</b>	별점과 리뷰를 작성할 수 있다면 다른 사용자들이 음식을 만들어볼지 말지 결정할 때 도움이 될 것이다.
<b>Approach</b>	조리법을 다 보고 난 후 별점과 리뷰를 등록할 수 있게 한다. 별점과 리뷰는 서버에 저장하며 다른 사람이 볼 땐 서버에서 데이터를 가져온다.
<b>Dynamics</b>	음식을 직접 만들어서 먹어본 후 조리법을 평가하고 싶은 경우
<b>Goals</b>	별점과 리뷰 기능을 구현한다.

## 5. Problem statement

이 장에서는 만들고자 하는 소프트웨어를 다양한 관점에서 생각한 결과를 기술한다. 가능한 한 root cause를 찾고자 했다. 첫 번째는 조리법의 다양성 문제이다. 앱의 기능이 좋다 하더라도 조리법이 몇 개 없다면 이 소프트웨어는 많은 사용자를 끌어들이지 못할 것이다. 관리자가 조리법을 많이 올리는 것으로 어느 정도 해소될 수 있지만 한 명이 올리는 것은 양적인 한계가 있다. 따라서 조리법을 등록하는 요리사들에 대한 만족할만한 보상을 줘서 조리법을 올리기 위한 동기부여를 확실하게 할 수 있는 방법을 찾는 것이 중요하다. 두 번째는 서버 관리와 API 사용에 대한 기술적 어려움이다. 앱을 제작하는 것 자체는 어느 정도 경험이 있지만 데이터베이스를 위한 서버 관리와 제품 정보를 얻기 위한 API 사용의 숙달이 되어 있지 않다. 따라서 많은 시간을 할애해서 소프트웨어가 적절히 동작하도록 주의를 기울여야 할 것이다. 세 번째는 음식 정보에 대한 저장 공간이다. 앱 출시와 동시에 제공되는 기본 조리법과 요리사가 등록한 조리법을 어디에 저장할지에 대한 문제이다. 이에 대해 일관성을 유지해서 모두 서버에 저장하는 것이 좋을 텐데 그렇다면 어떤 식으로 회원정보, 음식 정보 등 다양한 정보들을 저장할지 고민해야 한다. 이는 또한 두 번째 문제와도 연관이 있다. 네 번째는 저작권 문제이다. 다시 말하자면 관리자가 조리법을 등록할 때 음식 정보를 어디서 가져올 것인가 하는 문제 그리고 요리사가 등록한 조리법이 저작권에 위배되는지에 대한 문제이다. 다섯 번째는 요리사 자격 부여에 대한 명확한 자격 조건이다. 누구는 되고 누구는 안 되는 거에 대한 기준이 모호하다면 반발을 살 것이다. 여섯 번째는 취향별 음식 추천 알고리즘이다. 사용자가 기껏 답을 했는데 자신의 취향과 다른 음식이 나온다면 허탈할 것이다. 이 모든 것을 항상 염두에 두면서 설계를 이어나가야 할 것이다.

다음은 non functional requirement에 대한 내용이다. 후에 소프트웨어의 성능 요구 조건을 달성했는지 판단하기 위한 것을 목적으로 했다. 다음은 이 프로젝트의 NFS들이다.

- ① DB 접근 후 화면에 나타나는 시간은 3초 미만이어야 한다.
- ② 각 페이지는 2초 안에 로드되어야 한다.
- ③ 모든 조리법은 신뢰성 있는 출처를 가지고 있어야 한다.
- ④ 부드러운 화면 전환이 가능해야 한다.
- ⑤ 앱은 최소 안드로이드 Lollipop(API Level 21) 버전까지 지원해야 한다.

## 6. Glossary

이 장에선 보고서에 쓰인 용어들을 설명한다. 아래의 표를 통해 나타내었다.

[표 5-1] 용어 설명

용어	설명
전 세계의 음식들	이 프로젝트로 만들어지는 앱의 이름이다.
조리법	조리를 하는 방법 또는 기술이다. 페이지 별로 한 단계씩 보여준다.
관리자 / An administrator	앱의 운영을 관리하는 사람이다. 주로 조리법/요리사 등록 승인여부, 별점과 리뷰 검토 후 삭제를 담당한다.
요리사 / A Cook	일반 이용자 중에 신청과 검증을 통해 요리사가 된 이용자이다. 조리법을 적어 등록 요청할 수 있다.
이용자 / A user	일반 사용자이다. 할 수 있는 행동은 [표 2-1]을 참고하길 바란다.
네이버 REST API	음식의 재료 정보를 가져오기 위한 API이다.
DB 서버	회원 정보, 별점과 리뷰 등을 저장하기 위한 서버이다.

## 7. References

- 
- 1) Cooking at Home More During COVID-19? What to Know About Food Safety . (n.d.).  
<https://www.healthline.com/health-news/cooking-at-home-more-during-covid-19-what-to-know-about-food-safety>.