1. **User requirements definitions**
   1. **Objective**

User requirement definition에서는 사용자에게 어떤 서비스를 제공하는지에 대해서 설명한다. Functional requirement와 non-functional requirement로 분류해 서술하며, 이는 시스템을 구성하는 각 기능을 모두 포함한다. 자연어와 다이어그램을 활용하여 사용자의 이해를 돕고, 제품이나 프로세스가 따라야하는 표준이 있다면 이를 명시한다.

* 1. **Functional Requirements**

**A Sign up**

사용자에게 회원가입에 필요한 정보를 요청하는 기능이다. 사용자는 본 시스템의 보다 정확한 서비스를 제공받기 위해서는 회원가입을 해야 한다. 회원가입은 카카오 또는 네이버 계정의 연동을 통해 간편하게 가능하며, 사용자는 자신의 나이와 성별만을 입력하면 된다.

**B Log in**

사용자가 회원가입으로 생성된 아이디를 통해 시스템에 로그인하는 기능이다. 사용자는 시스템에 로그인하여, 시스템의 서비스를 제공받을 수 있다.

**C First Menu Evaluation**

사용자의 음식 취향을 분석하기 위한 기능이다. 본 시스템에 가입한 후 처음 로그인 하는 사용자는 먼저 자신에 취향에 맞게 시스템에서 제안된 100가지 음식을 평가해야 한다. 시스템은 평가 결과를 토대로 사용자의 취향을 분석하여 "음식추천"을 위한 모델을 생성한다.

**D Search**

사용자가 검색어를 통해 음식을 검색할 수 있는 기능이다. 사용자는 검색창에 음식의 이름을 입력하여 자신이 원하는 것을 손쉽게 찾을 수 있다. 이때 시스템은 사용자가 원하는 음식뿐만 아니라 입력된 단어를 토대로 비슷한 음식 또한 같이 보여준다.

**E Main Menu**

“First Menu Evaluation”에서 분석한 취향을 토대로 사용자에게 그들이 좋아할 만한 음식을 추천해주는 기능이다. 추천된 음식들이 나열되며, 그 음식을 만드는 데에 필요한 재료 또한 동시에 보여준다. 사용자는 이를 통해 자신이 원하는 음식을 선택할 수 있다.

**F Shopping Cart**

사용자가 “Main Menu”에서 마음에 든 음식을 선택할 경우 이 “장바구니(Shopping Cart)”에 담기게 된다. 사용자는 이 “장바구니”을 통해 자신이 어떤 음식을 선택했는지 한 눈에 확인할 수 있으며, 선택한 음식의 개수 및 가격 또한 확인이 가능하다. 만약 사용자가 선택한 음식을 취소하고 싶을 경우엔 ‘x’ 버튼을 눌러 쉽게 취소할 수 있다.

**G Order**

본 기능은 “Shopping Cart”의 연장선으로, 사용자는 자신의 장바구니의 담긴 음식을 “주문하기(Order)” 버튼 하나로 해결할 수 있다.

**H Post Evaluation**

사용자가 주문한 음식을 평가하는 기능이다. 사용자는 음식을 구입한 후에 마이페이지에서 “좋다”, “보통”, “싫다” 3가지 선택지를 기준으로 음식에 대한 평가를 남길 수 있다. 시스템은 이런 평가들을 통해 취향 분석 모델을 업데이트한다.

* 1. **Non-functional Requirements**

Non-functional requirement는 크게 product requirement, organization requirements, external requirement로 나눌 수 있다.

**A Product Requirements**

**A.1 Usability Requirements**

본 시스템은 추천 서비스이기에 앞서 음식을 주문하는 서비스이다. 따라서 수많은 음식 중에서 사용자가 원하는 음식을 찾기 쉬워야 하며, 음식에 대한 정보(필요한 재료, 가격 등)가 직관적이어야 한다.

**A.2 Performance Requirements**

본 시스템의 핵심 서비스는 “음식 추천”이다. 따라서 추천의 정확도가 낮으면 사용자의 이용율이 떨어질 수 있기 때문에 최대한 사용자의 취향에 근접해야 한다. 또한, 수많은 음식의 방대한 데이터를 나타내는 데에 문제가 없어야 하며, 음식을 검색했을 때 performance 측면에서 문제가 없도록 해야 한다.

**A.3 Dependability Requirement**

사용자에게 추천해준 음식이 그들의 취향에 맞지 않을 경우 사용자의 신뢰를 잃을 수 있으므로 최대한 많은 데이터를 모아 분석하여 취향에 근접한 결과를 내야만 한다. 또한, 주문/구매 서비스에 대해서도 오류가 발생할 경우 사용자의 신뢰를 잃을 우려가 있기 때문에, 이러한 방면에서도 문제를 만들어서는 안 된다.

**A.4 Security Requirement**

본 시스템의 경우 카카오 또는 네이버 계정을 연동하여 로그인하기 때문에 그에 대한 개인정보가 유출되어서는 안 되는 건 물론이고, 결제 수단에 대한 보안이 철저히 관리되어야 한다. 또한, 사용자가 시스템을 안심하고 이용할 수 있도록 데이터베이스 관리에 유의해야 한다.

**B Organizational Requirements**

**B.1 Environmental Requirements**

사용자가 언제 어디서든 사용할 수 있도록 모바일 디바이스는 상시 네트워크에 연결되어 있어야 한다. 또한, 데이터베이스를 관리하고 디바이스와 연결할 서버는 상시 구동되어야 한다.

**B.2 Operational Requirements**

시스템이 추천한 음식을 사용자가 마음에 들어 하지 않을 경우, 시스템은 계속해서 사용자의 요구에 맞는 음식을 추천할 필요가 있다. 따라서 사용자가 마음에 들어 하지 않는 데이터를 추가적으로 분석하여 요구에 근접한 결과를 도출하거나, 또는 사용자에게 보다 세부적인 질문을 하여 결과에 다가갈 수 있는 기능이 필요하다.

**B.3 Development Requirements**

보다 정확한 추천을 하기 위해 machine learning 기술, 더 나아가 deep learning 기술의 도입이 필요하다. 그리고 본 시스템은 안드로이드 기기를 대상으로 한 어플리케이션이기 때문에, 개발 언어는 Java를 사용해야 한다.

**C External Requirements**

**C.1 Regulatory Requirement**

본 시스템은 사용자에게 로그인을 요구하기에 앞서 먼저 그들의 개인정보를 수집함에 대한 동의를 구해야 하며, 취급방침에 대한 알림을 제공해야 한다. 또한, 정보를 외부에 제공하게 될 경우엔 사용자에게 동의를 먼저 구해야 한다.

**C.2 Safety/security Requirement**

본 시스템은 사용자의 개인정보가 외부로 유출되지 않도록 해야 하며, 사용자의 결제에 오류가 발생해서는 안 된다.

* 1. **Non-functional Requirements**

Non-functional requirement는 크게 product requirement, organization requirements, external requirement로 나눌 수 있다.

**A Product Requirements**

**A.1 Usability Requirements**

본 시스템은 추천 서비스이기에 앞서 음식을 주문하는 서비스이다. 따라서 수많은 음식 중에서 사용자가 원하는 음식을 찾기 쉬워야 하며, 음식에 대한 정보(필요한 재료, 가격 등)가 직관적이어야 한다.

**A.2 Performance Requirements**

본 시스템의 핵심 서비스는 “음식 추천”이다. 따라서 추천의 정확도가 낮으면 사용자의 이용율이 떨어질 수 있기 때문에 최대한 사용자의 취향에 근접해야 한다. 또한, 수많은 음식의 방대한 데이터를 나타내는 데에 문제가 없어야 하며, 음식을 검색했을 때 performance 측면에서 문제가 없도록 해야 한다.

**A.3 Dependability Requirement**

사용자에게 추천해준 음식이 그들의 취향에 맞지 않을 경우 사용자의 신뢰를 잃을 수 있으므로 최대한 많은 데이터를 모아 분석하여 취향에 근접한 결과를 내야만 한다. 또한, 주문/구매 서비스에 대해서도 오류가 발생할 경우 사용자의 신뢰를 잃을 우려가 있기 때문에, 이러한 방면에서도 문제를 만들어서는 안 된다.

**A.4 Security Requirement**

본 시스템의 경우 카카오 또는 네이버 계정을 연동하여 로그인하기 때문에 그에 대한 개인정보가 유출되어서는 안 되는 건 물론이고, 결제 수단에 대한 보안이 철저히 관리되어야 한다. 또한, 사용자가 시스템을 안심하고 이용할 수 있도록 데이터베이스 관리에 유의해야 한다.

**B Organizational Requirements**

**B.1 Environmental Requirements**

사용자가 언제 어디서든 사용할 수 있도록 모바일 디바이스는 상시 네트워크에 연결되어 있어야 한다. 또한, 데이터베이스를 관리하고 디바이스와 연결할 서버는 상시 구동되어야 한다.

**B.2 Operational Requirements**

시스템이 추천한 음식을 사용자가 마음에 들어 하지 않을 경우, 시스템은 계속해서 사용자의 요구에 맞는 음식을 추천할 필요가 있다. 따라서 사용자가 마음에 들어 하지 않는 데이터를 추가적으로 분석하여 요구에 근접한 결과를 도출하거나, 또는 사용자에게 보다 세부적인 질문을 하여 결과에 다가갈 수 있는 기능이 필요하다.

**B.3 Development Requirements**

보다 정확한 추천을 하기 위해 machine learning 기술, 더 나아가 deep learning 기술의 도입이 필요하다. 그리고 본 시스템은 안드로이드 기기를 대상으로 한 어플리케이션이기 때문에, 개발 언어는 Java를 사용해야 한다.

**C External Requirements**

**C.1 Regulatory Requirement**

본 시스템은 사용자에게 로그인을 요구하기에 앞서 먼저 그들의 개인정보를 수집함에 대한 동의를 구해야 하며, 취급방침에 대한 알림을 제공해야 한다. 또한, 정보를 외부에 제공하게 될 경우엔 사용자에게 동의를 먼저 구해야 한다.

**C.2 Safety/security Requirement**

본 시스템은 사용자의 개인정보가 외부로 유출되지 않도록 해야 하며, 사용자의 결제에 오류가 발생해서는 안 된다.

**Glossary**

사용자 – “eat it”의 서비스를 이용하는 사람을 말한다.

회원가입 – “eat it”의 회원이 되기 위해 특정 정보들을 입력하고 시스템에 등록하는 행위를 말한다.

로그인 - “eat it”에 등록한 정보를 통해 시스템에 접속하는 행위를 말한다.

카카오 or 네이버 계정 연동 – 카카오톡에서 사용하는 ID를 “eat it”에서도 사용할 수 있게 카카오톡 서비스를 연결시키는 것을 말한다.

검색 – 사용자가 입력한 단어에 적합한 정보를 찾아 보여주는 기능을 말한다.

마이페이지- 사용자가 음식 구입 후에 음식에 대한 평가를 남길 때 사용하는 페이지를 말한다.

업데이트 – 기존의 정보나 데이터를 수정, 변경하거나 최신의 정보나 데이터를 추가하는 일을 말한다.

서버 – 서비스에 응답하는 프로세스와 하드웨어를 통괄하는 용어를 말한다.

데이터베이스 - 서버에서 서비스를 제공할 때, 필요한 정보들을 저장하고 있는 저장소를 말한다.

Machine learning – 컴퓨터가 학습할 수 있도록 하는 알고리즘과 기술을 개발하는 분야를 말한다.

Deep learning – 사물이나 데이터를 군집화 하거나 분류하는 데 사용되는 일종의 기술적 방법론을 말한다.