

Requirement Specification

Wizard of Order



제출일	2018.05.06	그룹	Group 6
과목	소프트웨어공학개론	담당교수	이은석 교수님
이름	김윤성	학번	2015312606
이름	박종원	학번	2014314516
이름	이지훈	학번	2016314088
이름	이혜리	학번	2014314853
이름	조건봉	학번	2016313167

Contents

1. Preface	6
1.1. Objective	6
1.2. Readership	6
A. User Requirement Readership	6
B. System Requirement Readership	6
1.3. Document Structure	6
A. Preface	6
B. Introduction	7
C. Glossary	7
D. User Requirement Definition	7
E. System Architecture	7
F. System Requirement Specification	7
G. System Model	8
H. System Evolution	8
I. Appendices	8
J. Index	8
2. Introduction	9
2.1. Objective	9
2.2. Needs	9
2.3. Wizard of Order	11
2.4. Expectations	12
A. Restaurant	12
B. Customer	12

3.	Glossary.....	13
3.1.	Objective	13
3.2.	Term definition	13
3.3.	Acronyms and Abbreviations.....	14
4.	User Requirement Definition.....	15
4.1.	Objective	15
4.2.	Functional Requirements	15
A.	Set NFC Tag.....	15
B.	Load Webpage.....	15
C.	Select Menu	15
D.	Confirm Order	16
E.	Submit Order	16
F.	Final Page	16
G.	Manage Orders.....	16
4.3.	Non-functional Requirements	17
A.	Product Requirement.....	17
B.	Organization Requirement	18
C.	External Requirement.....	18
5.	System Architecture.....	19
5.1.	Objective	19
5.2.	Order Create System	20
5.3.	Order Management System	21
5.4.	Restaurant System	22
6.	System Requirement Definition	23

6.1.	Objective	23
6.2.	Functional Requirement	23
A.	Set NFC Tag.....	23
B.	Load Webpage.....	24
C.	Select Menu	25
D.	Confirm Order	25
E.	Submit Order	26
F.	Final Page	26
G.	Manage Orders.....	27
6.3.	Non-functional Requirement	27
A.	Product Requirement	27
B.	Organization Requirement	28
C.	External Requirement.....	29
7.	System Models.....	36
7.1.	Objective	36
7.2.	Context Models.....	36
A.	Context Model	36
B.	Process Diagram	37
7.3.	Interaction Models.....	38
A.	Use Case Models	38
B.	Tabular description for each use case'	38
C.	Sequence Diagram.....	43
7.4.	Structural Models	44
A.	Class Diagram.....	44

7.5.	Behavioral Models	44
A.	Data-Driven Modelling.....오류! 책갈피가 정의되어 있지 않습니다.	
B.	Event-Driven Modelling	44
8.	System Evolution.....	45
8.1.	Objective	45
9.	Appendices.....	49
9.1.	Objective	49
9.2.	Database Requirements	49
9.3.	User System Requirements.....오류! 책갈피가 정의되어 있지 않습니다.	
9.4.	Development Process.....오류! 책갈피가 정의되어 있지 않습니다.	
10.	Index	51
10.1.	Table Index.....	51
10.2.	Figure Index	51
10.3.	Diagram Index.....	52

1. Preface

1.1. Objective

Preface에서는 본 문서의 독자를 정의하고, 문서의 전반적인 구성과 각 파트에서 서술하는 내용에 대해 설명한다. 또한, 문서 내용의 갱신에 따른 변경사항과 그 사유에 대해 표 형식으로 기술한다.

1.2. Readership

A. User Requirement Readership

User requirement에서는 본 software system의 요구사항에 대한 을 사용자의 관점에서 추상화 및 시각화를 통해 정의하고, 이를 서술한다. 전문지식이 없는 독자를 고려하여, user requirement는 자연어와 도표 및 그림 등을 이용해 설명한다. Client manager와 contractor manager가 주된 독자가 된다.

B. System Requirement Readership

System requirement는 시스템을 구성하는 function과 service 그리고 operational constraint에 대해 상세히 기술한다. 이는 고객과의 계약에 중요한 부분이 될 수 있으므로 구조화된 문서 양식을 채택하여 설명한다. 시스템을 개발하는 software developer와 system architects를 비롯한 개발진이 주된 독자가 된다.

1.3. Document Structure

A. Preface

Preface에서는 예상 독자를 정의하고, 문서의 전반적인 구성과 각 파트에서 서술하는 내용에 대해 설명합니다. 또한, 문서 내용의 갱신에 따른 변경사항과 그 사유에 대해 표 형식으로 기술합니다.

B. Introduction

Introduction에서는 본 시스템의 필요성을 실제 사회에서의 needs를 분석하고, 기존 유사 제품의 한계점 및 개선점과 같이 설명합니다. 본 시스템의 목적과 이점을 다른 서브 시스템과 어떻게 동작하는지를 통해 간략히 기술합니다.

C. Glossary

Glossary에서는 본 문서에서 나오는 용어들의 뜻을 설명합니다. 배경지식이 없는 독자를 고려하여, 문서를 이해하는데 어려움이 없도록 가능한 모든 용어에 대해 서술합니다. 단어의 명확한 정의와 기술을 원칙으로 하여, 뜻을 함부로 추정하지 않도록 합니다.

D. User Requirement Definition

User requirement definition에서는 본 시스템에서 제공하는 서비스와 제약사항들에 대해 사용자 관점을 중심으로 설명합니다. 따라서 이를 Functional requirement와 non-functional requirement로 분류해 기술할 수 있으며, 이는 시스템을 구성하는 각 기능을 모두 포함합니다. 자연어와 다이어그램이 활용하여 사용자의 이해를 돕고, 제품이나 프로세스가 따라야 하는 표준이 있다면 이를 명시합니다.

E. System Architecture

System architecture에서는 본 시스템을 구성하는 architecture 개요를 고수준으로 설명합니다. 시스템을 구성하는 컴포넌트를 명시하고, 이들 간의 상호작용을 시각화를 통해 그림으로 제시합니다. 이는 개발자가 시스템의 전체적 분포를 이해하는데 도움을 줍니다.

F. System Requirement Specification

System requirement specification에서는 시스템을 구성하는 functional requirement 그리고 non-functional requirement에 대해 기술적인 관점에서 기능적인 부분을 더 상세히 기술한다. 기본적으로 구조화된 문서 양식을 통해 시스템 내 기능적인 부분을 명확히 설명한다. 그리고 non-functional requirement의 경우는 필요에 따라, 다이어그램 및 그림을 이용한

자연어를 통해 최대한 자세히 설명하도록 한다.

G. System Model

System model에서는 시스템을 구성하는 서브 시스템, 그리고 컴포넌트들의 상호작용과 그 관계를 설명한다. Object model, data-flow model 그리고 semantic data model 등을 활용해 작성한다.

H. System Evolution

System evolution에서는 시스템을 위한 주요 가정들에 대해 설명하고, 발생할 수 있는 변화를 예측한다. 이는 하드웨어의 발전, 사용자의 요구사항 변화 등에 기반한다. 나아가 어떻게 이러한 변화사항을 어떻게 반영할 수 있는지를 살펴본다. 이는 시스템 설계자가 추후에 시스템의 변경을 피할 수 있도록 해주고 변화에 적응가능한 시스템을 개발할 수 있도록 도와준다.

I. Appendices

Appendices에서는 개발되는 시스템에 대한 구체적이고 자세한 정보를 제공한다.

J. Index

Index에서는 본 문서에서 사용된 용어와, 다이어그램, 기능에 대한 인덱스를 나타낸다.

2. Introduction

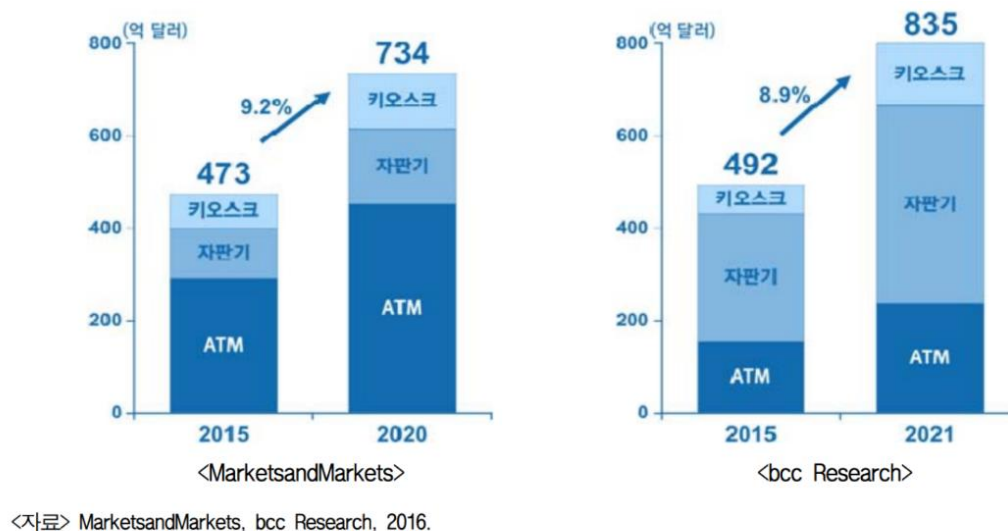
2.1. Objective

Preface에서는 예상 독자를 정의하고, 문서의 전반적인 구성과 각 파트에서 서술하는 내용에 대해 설명합니다. 또한, 문서 내용의 갱신에 따른 변경사항과 그 사유에 대해 표 형식으로 기술합니다.

2.2. Needs

최근, 식당들에 키오스크 (kiosk; 무인정보단말기)의 설치가 확대되고 있다. 특히 델라코트, 아워홈 같은 대형 푸드 코트나 롯데리아, 맥도날드 등 패스트푸드 매장에서 적극적으로 키오스크를 도입하는 추세인데, 많은 주문이 들어오는 식당에서 메뉴 주문과 결제를 무인화, 자동화함으로써 손님이 붐비는 상황에서 종업원의 부담을 줄여 더 쾌적한 서비스를 제공할 수 있다. (김용균, 2017)

(Researchbbc, 2016)



[그림 1] 세계 대화형 키오스크 시장 전망

Figure 1. Kiosk Market Outlook

그러나 분업화되어 음식이 바로바로 나오고 기다리는 줄, 버리는 곳이 정해져 있는 대규모

식당과는 다르게, 소형 식당은 키오스크 도입 시 공간이 좁은 경우 잦은 이동으로 인한 번거로움, 대기지연 및 동선의 비효율성, 다양한 음식들의 조리 시간이 다름에도 번호표 기반의 음식 수령으로 인한 불편함이 생긴다. 즉, 기존 식당에서는 식당 내 구조적 문제 발생으로 인해 키오스크를 도입하기 어려웠다.



Figure 2. Tablepad example

따라서, 키오스크와는 다른 주문 자동화 기기가 필요하다. 기기는 주문을 위한 자리 이탈 및 다른 손님들의 주문을 기다리지 않게 하고, 동선과 가게의 미관을 해치는 키오스크 대기 줄을 없애고, 너댓 명 남짓의 단체 손님의 주문을 편리하게 해야 한다. 시장에 나와 있는 비슷한 제품으로는 테이블 패드가 있는데, 이는 초기 설치 비용이 많이 들고 비교적 큰 부피를 차지한다. 이런 점들을 고려해, 소규모 식당도 사용할 수 있는 새로운 주문 자동화 시스템을 만드는 것이 우리의 목표이다.

2.3. Wizard of Order

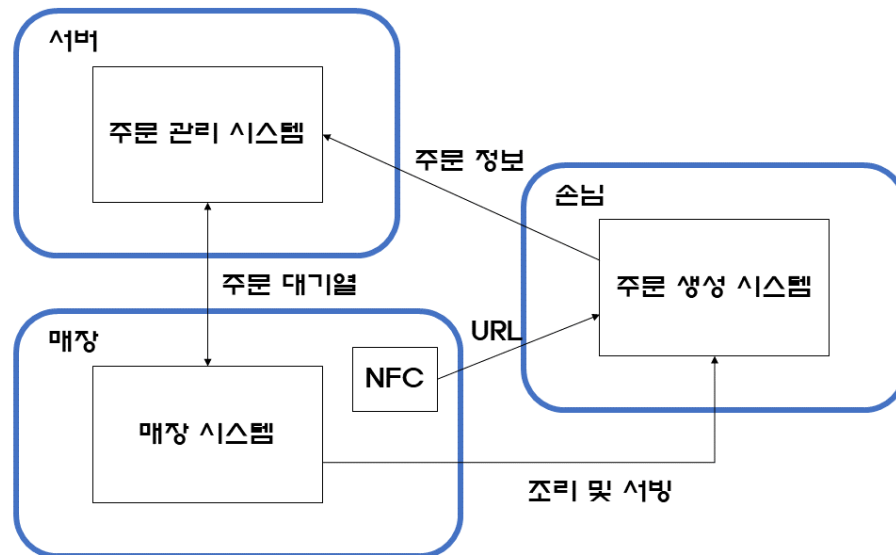


Diagram 1. Entire System

오더의 마법사는 식당 내 테이블마다 부착된 NFC 태그를 이용해 손님이 쉽게 웹에서 메뉴 확인 및 주문을 할 수 있게 하고, 매장에서는 종업원이나 손님이 왔다 갔다 할 필요 없이 테이블 별로 주문을 관리할 수 있게 한다.

시스템은 크게 세부분으로 구성되는데, 손님이 NFC를 통해 접근한 웹으로 제공되는 주문 시스템, 손님이 요청한 주문을 기록하고 매장에 알려주는 주문 관리 시스템, 매장에서 현재 대기중인 주문을 확인하고 관리하는 대기 주문 시스템이다.

주문 시스템은 손님의 스마트폰에서 웹을 통해 접근(?)되며, 세 개의 서브시스템으로 나뉜다. 손님에게 메뉴를 보여주고 주문을 고르게 하는 메뉴 선택 시스템, 주문을 확정하고 음료 수나 고명, 주방 전달 사항들을 정해 주문 관리 시스템에 전달하는 주문 확정 시스템, 주문 완료 후 주문 접수 확인 메시지나 감사 인사, 광고 등을 전달하는 주문 마침 시스템이 있다.

주문 관리 시스템은 중앙 서버에서 작동하며, 두 개의 서브시스템으로 나뉜다. 주문 저장 시스템은 손님이 보내온 주문 정보를 받아 추후 분석을 위해 저장하고, 대기 주문 시스템 또한 주문 정보를 이용해 매장에서 필요한 정보 (현재 주방에 밀려 있는 주문 목록 등)을 관리하고 매장 시스템과 상호작용한다.

매장 시스템은 매장에 설치된 단말기나 종업원들의 스마트폰에서 웹을 통해 접근(?)되며, 두

개의 서브시스템으로 나뉜다. 주방에서 현재 대기 중인 (조리해야 하는) 주문을 대기 주문 시스템에서 받아와 먼저 들어온 주문 순으로 출력하고, 조리가 끝난 주문을 대기열에서 지우는 대기열 수정을 하는 조리 대기열 시스템, 서빙을 맡은 홀 종업원이 들어온 주문을 확인해 기존 시스템에 입력하고, 조리가 끝난 주문을 어디에 서빙 해야 할 지 알려주는 홀 시스템이 있다.

2.4. Expectations

A. Restaurant

손님이 붐비는 상황에서 종업원의 업무 부담이 과하던 상황을 주문 자동화를 통해 해결할 수 있다. 키오스크와 같은 기존의 시스템은 주문에 시간이 오래 걸리기 때문에 손님이 붐비는 상황에서는 자리를 비운 채 키오스크에 줄을 길게 서게 되고, 가게의 미관에도 영향을 줄 뿐 더러 사람들의 동선도 해쳤지만, 오즈의 마법사는 손님이 테이블에 앉아 개인 스마트폰으로 주문하기 때문에 줄을 서서 기다릴 필요도 없고, 주문을 받아야 했던 종업원의 부담도 줄여준다. 또한 모든 주문이 접근하기 편하게 저장되기 때문에, 추후 통계적 분석에 유용하게 사용할 수 있다.

B. Customer

기존의 키오스크를 통한 주문이나 종업원을 통한 주문은 차례를 기다려야 했지만, 오즈의 마법사는 손님이 테이블에서 웹을 통해 바로 주문을 하기 때문에 기다릴 필요가 없다. 또한 주문을 위해 이동해 줄을 서거나 종업원과 대면할 필요가 없다.

3. Glossary

3.1. Objective

Glossary에서는 본 문서에서 나오는 용어들의 뜻을 설명합니다. 배경지식이 없는 독자를 고려하여, 문서를 이해하는데 어려움이 없도록 가능한 모든 용어에 대해 서술합니다. 단어의 명확한 정의와 기술을 원칙으로 하여, 뜻을 함부로 추정하지 않도록 합니다.

3.2. Term definition

Table 1 Term definition

용어	정의
데이터(data)	자료, 자재라는 뜻. 설계, 계획 등에 필요한 자료, 즉 공식, 계산 도표, 치수표, 조사 지표 등을 일컫는 말
자동화 시스템	운영자는 관리자와 관찰자 뿐이므로 시스템 작동에 사람의 통제가 필요하지 않음
Python	오픈소스 고급 프로그래밍 언어 중 하나이다. C언어를 기반으로 한 오픈소스 고급 프로그래밍 언어
Django 프레임워크	Django는 Python으로 작성된 오픈 소스 웹 애플리케이션 프레임 워크 소프트웨어 어플리케이션이나 솔루션의 개발을 수월하게 하기 위해 소프트웨어의 구체적 기능들에 해당하는 부분의 설계와 구현을 재사용 가능하도록 협업화된 형태로 제공하는 소프트웨어 환경을 말한다. 프레임워크 [Framework]
웹(Web)	인터넷에 연결된 컴퓨터들을 통해 사람들이 정보를 공유할 수 있는 전 세계적인 정보 공간을 말한다. 월드 와이드 웹(WWW)이라 부르는 경우가 많다.
System Model	시스템을 구성하는 서브 시스템, 그리고 컴포넌트들의 상호작용과 그 관계를 설명한다.
알촌(Alchon)	학교 근처에 자동의 식사 주문 기계 있는 식당.
단말기	디지털 자료 전송 시스템에서 자료를 만들거나 보기 위한 기기. 또는 자료를 보내거나 받기 위한 기능을 수행하는 기기로서 사람과 직접 대면하게 되는 자료 처리의 기본 기기.
어플리케이션 (APP)	스마트폰 안에서 구동되는 프로그램.
키오스크 (KIOSK)	무인 정보 단말기: 공공장소에 설치된 터치스크린 방식의 정보전달 시스템.
업데이트	어떤 것에 대한 추가, 삭제, 수정 등의 이동데이터에 의해서 마스터 파일의 내

(Update)	용을 고쳐 쓰고 파일의 내용을 현상태로 유지하는 것을 말한다.
클라이언트 (Client)	클라이언트/서버(client/server) 구성에서 사용자측. 사용자가 서버에 접속했을 때 클라이언트는 사용자 자신을 지칭할 수도 있고, 사용자의 컴퓨터를 가리키기도 하며, 컴퓨터에서 동작하고 있는 프로그램이 될 수도 있다. 클라이언트 [client] (컴퓨터인터넷IT용어대사전, 2011. 1. 20., 일진사) 클라이언트 [client] (컴퓨터인터넷IT용어대사전, 2011. 1. 20., 일진사)
데이터베이스 (Database)	여러 사람에 의해 공유되어 사용될 목적으로 통합하여 관리되는 데이터의 집합을 말한다. 자료항목의 중복을 없애고 자료를 구조화하여 저장함으로써 자료 검색과 갱신의 효율을 높인다.
빅 데이터 (Big data)	빅 데이터란 디지털 환경에서 생성되는 데이터로 그 규모가 방대하고, 생성 주기도 짧고, 형태도 수치 데이터뿐 아니라 문자와 영상 데이터를 포함하는 대규모 데이터를 말한다.
마케팅 (Marketing)	생산자가 상품 또는 서비스를 소비자에게 유통시키는 데 관련된 모든 체계적 경영활동.
솔루션 (Solution)	어떤 특정한 상황에 대한 해결책으로 사용자의 요구에 따라 프로그램과 관련된 문제들을 처리해주는 하드웨어, 소프트웨어, 기술 등을 가리킨다.

3.3. Acronyms and Abbreviations

Table 2 Acronyms and Abbreviations

용어	정의
URL (Uniform Resource Locator)	Uniform Resource Locator , 웹 사이트에 들어가기 위해 필요한 주소.
NFC (Near Field Communication)	가까운 거리에서 다양한 무선 데이터를 주고받는 통신 기술이다.
GPS (Global Positioning System)	GPS(Global Positioning System)는 GPS 위성에서 보내는 신호를 수신해 사용자의 현재 위치를 계산하는 위성 항법 시스템이다.
UI (User Interface)	사용자 인터페이스 : 휴대폰, 컴퓨터, 내비게이션 등 디지털 기기를 작동시키는 명령어나 기법을 포함하는 사용자 환경을 뜻한다. 스마트폰의 경우 애플리케이션 아이콘 형태 및 화면 구성을 가리킬 때가 많다.
UX (User eXperience)	사용자 경험: 사용자가 어떤 시스템, 제품, 서비스를 직/간접적으로 이용하면서 느끼고 생각하게 되는 총체적 경험을 말한다.

4. User Requirement Definition

4.1. Objective

User requirement definition에서는 본 시스템에서 제공하는 서비스와 제약사항들에 대해 사용자 관점을 중심으로 설명합니다. 따라서 이를 Functional requirement와 non-functional requirement로 분류해 기술할 수 있으며, 이는 시스템을 구성하는 각 기능을 모두 포함합니다. 자연어와 다이어그램이 활용하여 사용자의 이해를 돕고, 제품이나 프로세스가 따라야하는 표준이 있다면 이를 명시합니다.

4.2. Functional Requirements

A. Set NFC Tag

사용자 단말기로 웹 사이트를 이용할 수 있도록 NFC 태그에 해당 웹 URL을 인코딩한다. 주문 자동화 시스템을 어플리케이션이 아닌 웹 서비스로 제공함으로써 사용자가 주문을 위해 따로 어플을 다운받는 번거로움을 줄여준다. 테이블마다 URL 내 고유 id를 부여하여 이것으로 테이블 번호를 구별할 수 있다.

B. Load Webpage

단말기에 URL 정보를 주어 웹 서비스로 연결해주는 기능이다. 사용자는 맨 처음 NFC 태그로 웹 페이지에 접속한 후, 바로 식당 메뉴판을 보며 음식을 고르고 장바구니에 담을 수 있다. 또한 메뉴 고르기를 비롯해 주문 수정, 최종 주문 과정 등 본 서비스의 모든 기능은 웹 페이지를 불러오는 방식을 취한다. 이때, URL 쿼리에 사용자가 주문할 음식의 정보를 담아 여러 페이지를 옮겨 다닐 때도 주문 정보를 유지하도록 한다.

C. Select Menu

식당 메뉴 페이지를 통해 주문할 음식들을 고르는 기능이다. 사용자는 자리에 앉아 자신의 단말기로 메뉴를 고르고 이를 장바구니에 담는다. 음식은 기본적으로 종류에 따른 카테고리 정렬되어 있고, 사용자는 '인기도', '가격' 등의 속성 순으로 정렬할 수 있게 한다. 식당의 요구사항에 맞춰 '이벤트 페이지', '이달의 신메뉴' 등 다양한 방식으로 음식을 홍보할 수 있게

한다. 또한, 사용자가 메뉴를 담을 때는 수량을 조정하고 음식에 따라서 추가 옵션을 선택할 수 있도록 한다.

D. Confirm Order

Confirm order는 자신이 음식을 담은 장바구니를 확인하고 이를 수정하는 기능을 제공한다. 여기서 사용자는 최종 주문을 하기 전에 주문서를 확인 및 조정하는 과정을 거친다. 수량 조정 및 메뉴 삭제는 해당 페이지에서 바로 처리할 수 있고, 새로운 메뉴를 추가하는 것은 '뒤로 가기' 기능을 제공하여 장바구니 정보를 유지한 채 select order 부분으로 넘어갈 수 있게 한다. '최종 주문' 버튼을 제공하여 음식을 주문할 수 있다.

E. Submit Order

Submit order는 사용자가 '최종 주문' 버튼을 눌렀을 때, 주문서 정보를 음식점 단말기에 전송하여, 주방과 직원이 이를 확인하고 음식주문이 최종적으로 들어갈 수 있게 한다. 또한 주문 정보를 웹 서버의 주문서 데이터베이스에 기록한다. 이는 추후에 빅데이터로 활용되어 손님들의 주문 양상과 메뉴들의 판매량을 측정할 수 있게 해준다.

F. Final Page

Final page에서 사용자는 자신의 주문이 완료되었음을 확인할 수 있다. '추가 주문' 버튼을 제공하여 다시 주문 페이지로 돌아가 주문을 추가할 수 있도록 한다.

G. Manage Orders

Manage order는 식당 직원이 음식점 단말기에서 손님들의 주문서 정보를 주문 순서대로, 혹은 테이블별로 확인하고 이를 조정할 수 있는 기능을 제공한다. 본 기능은 하나의 페이지에서 실시간으로 가능해야 한다. 혹시 사용자가 주문을 취소하거나, 서빙이 완료되었을 때, 해당 주문서를 삭제하거나 조정할 수 있는 인터페이스를 제공한다.

4.3. Non-functional Requirements

A. Product Requirement

A.1. Performance Requirement

기본적으로 본 시스템은 사용자의 단말기에서 주문을 하는 웹 서비스이다. 만약 일부 단말기에서 웹 페이지가 제대로 보이지 않거나 데이터 처리 속도가 느리다면 문제가 발생할 것이다 따라서, 이러한 호환성을 높여 가능한 모든 단말기에 웹 페이지 제대로 보여지고 제대로 된 처리속도를 보장할 수 있도록 한다.

A.2. Security Requirement

본 시스템의 웹 페이지는 모두 NFC 태그를 통해서만 접근해야 한다. 따라서, 외부에서 URL에 접속해 의도적으로 주문을 하지 못하도록 해야 한다. 또한, 지속적으로 쌓이는 식당의 주문 내역 혹은 음식 데이터베이스 내 정보가 유출되지 않도록 한다.

A.3. Efficiency Requirement

본 시스템은 동시 접속자가 한 식당내 손님들로 한정적이다. 따라서, 큰 규모의 서버를 요구하지 않지만 시스템 performance에 악영향을 주지 않도록 적절한 규모의 서버를 구축한다.

A.4. Dependability Requirement

사용자는 시스템 상에서 웹 페이지를 옮겨 다닐 때, 자신의 장바구니 정보를 URL 쿼리에 담아 일관성을 유지한다. 따라서, 이 정보가 시스템 오류로 변경되거나 손상되지 않도록 한다.

A.5. Usability Requirement

식당에서 사용하는 음식 주문 서비스인만큼, 웹 서비스 이용의 편의성을 최대한 높여야 한다. 따라서, 최종 주문 단계까지의 과정을 최소화하고 인터페이스를 직관적으로 설계해야 한다.

B. Organization Requirement

B.1. Environmental Requirement

기본적으로 식당 테이블에 NFC 태그가 부착이 되어있어야 하며 이것이 물리적으로 손상되거나 URL 인코딩 정보가 잘못되어서는 안된다. 또한, 사용자 단말기를 통한 웹 서비스로 주문을 하기 때문에, 손님들 중 한명은 꼭 웹 사이트 접속이 가능한 단말기를 지니고 있어야 한다.

B.2. Operational Requirement

본 시스템은 음식점에서도 손님들의 주문 내역을 확인하고 이를 조정할 수 있는 인터페이스를 제공해야 한다. 손님의 입장에서, 정확한 음식 주문을 위해 자신의 주문 내역을 확인하고 수정할 수 있도록 한다.

B.3. Development Requirement

사용자의 주문 내역 정보를 URL 쿼리에 저장하여 주문 과정을 거치는 동안 여러 정적 웹 페이지를 이동하는 동안 정보가 기억될 수 있도록 한다. 단말기의 종류에 영향을 덜 받는 웹 서비스를 채택하고, Django 프레임워크를 통해 하드웨어와 웹 서비스 간의 호환성을 높인다.

C. External Requirement

C.1. Legislative Requirement

축적되는 식당의 주문 내역 데이터를 빅데이터 분석에 활용하기 위해, 식당의 주문 내역에 접근하고 이를 이용할 수 있도록 식당 측의 동의를 얻어야 한다.

C.2. Safety/Security Requirement

본 시스템은 식당의 모든 음식 판매 및 주문 내역 정보를 데이터베이스로 관리하기 때문에, 정보 유출을 방지하기 위해 보안성을 높여야 한다. 또한 NFC에 인코딩한 URL이 외부에서 접근되지 않도록 한다.

5. System Architecture

5.1. Objective

System architecture에서는 본 시스템을 구성하는 architecture 개요를 고수준으로 설명합니다. 시스템을 구성하는 컴포넌트를 명시하고, 이들 간의 상호작용을 시각화를 통해 그림으로 제시합니다. 이는 개발자가 시스템의 전체적 분포를 이해하는데 도움을 줍니다.

5.2. Overall Architecture

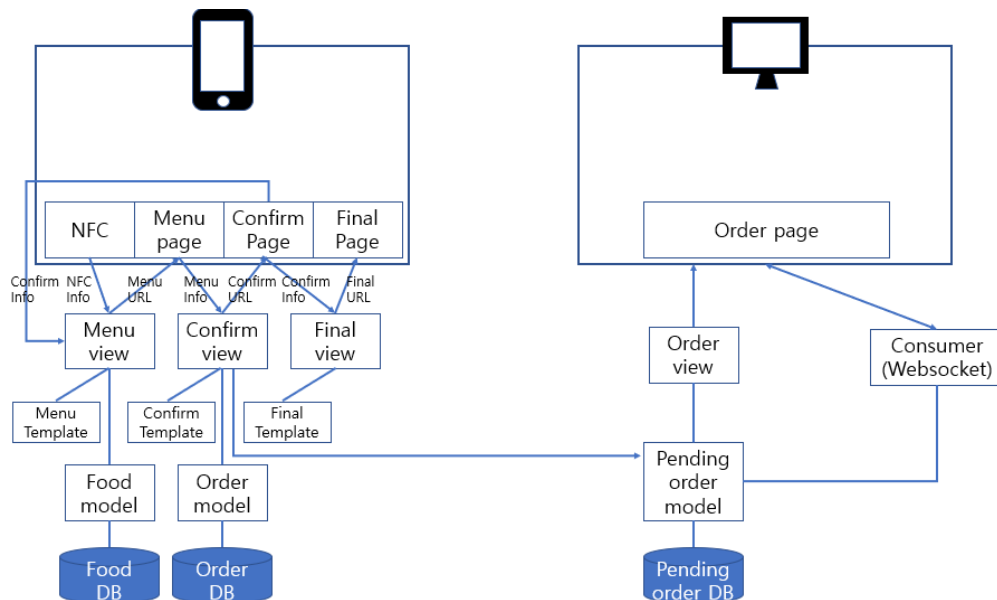


Diagram 2. Architectural Diagram

5.3. Order Create System

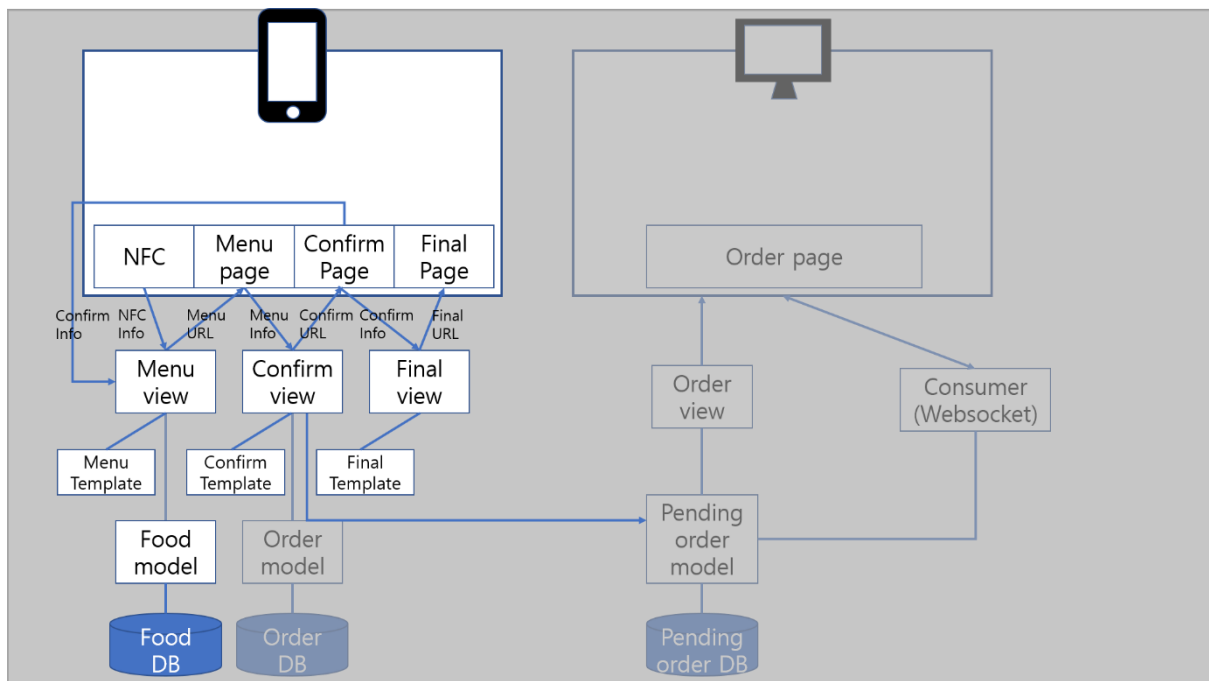


Diagram 3. Architectural Diagram – Order Create System

주문 생성 시스템은 손님이 메뉴를 전달받아 주문을 정하고 매장 서버로 보내는 시스템이다. 메뉴 선택 시스템에서는 메뉴 데이터베이스의 메뉴 정보가 손님에게 제공되고, 손님은 메뉴 선택 페이지를 통해 주문을 생성한다. 주문이 만들어지면 주문 확정 시스템에서 수량과 종류를 확인하고 수정하는 주문 확정을 한다. 주문이 확정되면 주문 마감 시스템에서 주문 데이터가 주문 관리 시스템에 전달되고 들어간 주문을 확인하는 최종(?) 페이지를 띄운다.

5.4. Order Management System

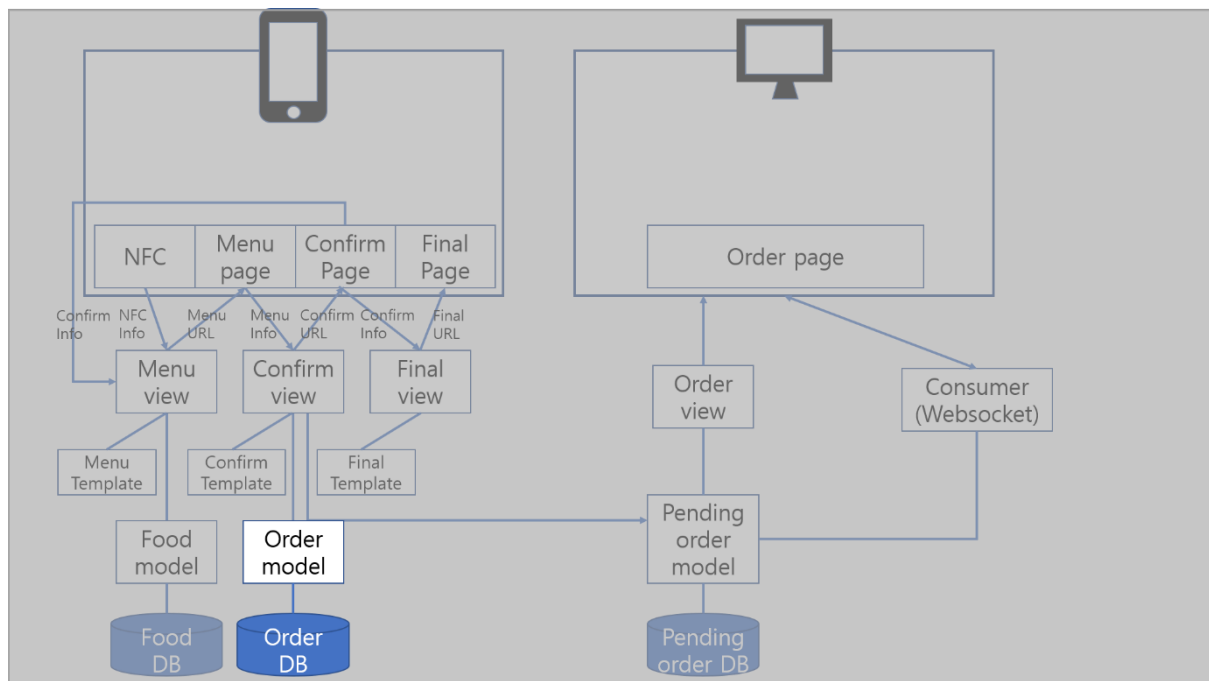


Diagram 4. Architectural Diagram – Order Management System

주문 관리 시스템에서는 손님들이 주문한 정보를 주문 확정 시스템에서 받아와 주문 저장 시스템이 추후 분석을 위해 데이터베이스에 저장하고, 또 대기 주문 시스템이 매장에서 현재 대기 중인 주문을 DB와 대기열에 저장한다. 대기 주문 시스템은 대기열 정보와 주문 정보를 매장 시스템에 제공하고, 매장 시스템에서 보내온 주문 진행 상황 정보를 통해 대기 주문 상태를 업데이트 한다.

5.5. Restaurant System

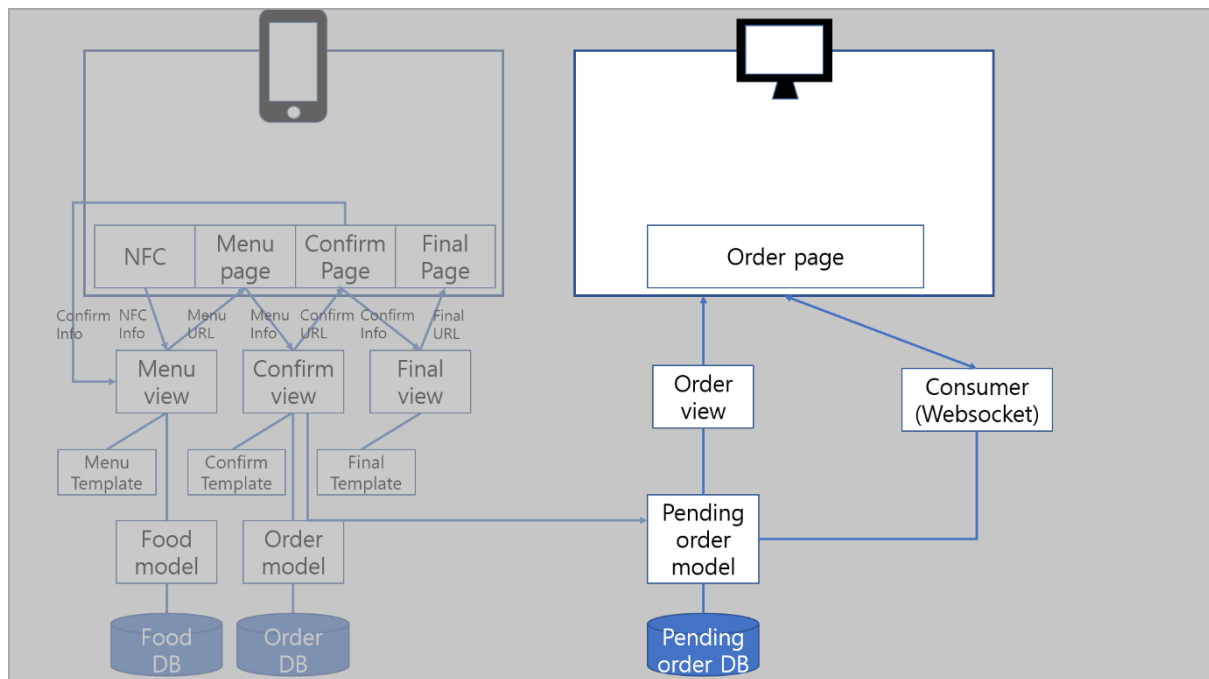


Diagram 5. Architectural Diagram – Restaurant System

매장 시스템에서는 조리 대기열 시스템이 현재 조리를 해야 하는 조리 대기열을 대기 주문 시스템에서 읽어오고 조리가 끝난 주문을 대기열에서 삭제하고, 홀 시스템이 현재 테이블 별 주문 상황과 주방에서 조리가 끝난 주문을 어디로 서빙 해야 할 지를 알려준다. (보완하거나 주문 관리 시스템이랑 합쳐야 할 듯)

6. System Requirement Definition

6.1. Objective

System requirement specification에서는 시스템을 구성하는 functional requirement 그리고 non-functional requirement에 대해 기술적인 관점에서 기능적인 부분을 더 상세히 기술한다. 기본적으로 구조화된 문서 양식을 통해 시스템 내 기능적인 부분을 명확히 설명한다. 그리고 non-functional requirement의 경우는 필요에 따라, 다이어그램 및 그림을 이용한 자연어를 통해 최대한 자세히 설명하도록 한다.

6.2. Functional Requirement

A. Set NFC Tag

Table 3. Functional Requirement – Set NFC Tag

Functional Requirement	
기능	Set NFC Tag
설명	NFC Tools 어플리케이션을 사용하여 NFC 태그에 본 서비스 웹 URL을 인코딩한 후 테이블에 부착시킨다. 이때, 각 테이블마다 다른 고유번호를 가지게끔 한다.
입력	URL 정보 + 테이블 고유번호
출력	-
처리	
조건	-

B. Load Webpage**Table 4. Functional Requirement – Load Webpage**

Functional Requirement	
기능	Load Webpage
설명	스마트폰에 인코딩된 NFC를 인식시켰을 때 자동으로 웹 브라우저가 켜지고 해당 URL이 자동으로 입력될 수 있도록 한다. 해당 URL은 본 시스템 서버의 IP주소를 받고 서버에서 메뉴판 정보를 단말기에 보내준다.
입력	URL 정보
출력	메뉴판 정보
처리	URL로 본 서버의 IP 주소를 얻고 메뉴판 정보를 요청한다. 서버로부터 받은 메뉴판 정보를 단말기에 띄워준다.
조건	NFC에 스마트폰을 가져다 대었을 때

C. Select Menu

Table 5. Functional Requirement – Select Menu

Functional Requirement	
기능	Select Menu
설명	메뉴 페이지에서 본인이 원하는 메뉴를 선택하여 장바구니에 담는 기능이다. 메뉴 페이지는 상단에서부터 크게 음식점 정보, 메뉴 정보, 장바구니 담기 버튼으로 이루어져 있다. 음식점 정보는 음식점 사진, 이름 등으로 구성되어 있다. 메뉴 정보는 해당 음식점에서 제공하는 모든 메뉴들의 정보를 포함한다. 이때, 사용자가 원하는 메뉴를 클릭하면 해당 메뉴 이미지에 음영표시가 되어 선택되어 있음을 표시한다. 장바구니 담기 버튼은 선택된 메뉴가 없으면 비활성 되어있다가 선택된 메뉴가 있으면 활성 상태로 변한다. 비활성과 활성 버튼은 색깔로 차이를 준다. 메뉴를 선택 후, 장바구니 담기 버튼을 누르면 해당 메뉴 정보를 메뉴확인 페이지로 전달한다.
입력	메뉴 선택, 장바구니 담기 버튼 클릭
출력	선택된 메뉴의 정보
처리	
조건	

D. Confirm Order

Table 6. Functional Requirement – Confirm Order

Functional Requirement	
기능	Confirm order
설명	메뉴 확인 페이지에서 본인이 선택한 메뉴들을 확인하고 수정할 내용이 있으면 뒤로가기 버튼을 클릭한다.
입력	뒤로가기 버튼
출력	메뉴 페이지(선택된 메뉴 정보와 함께)
처리	뒤로가기 버튼을 클릭 시, 장바구니에 있던 메뉴 정보를 메뉴 페이지에 보내 미리 선택되어 있게끔 한다.
조건	-

E. Submit Order

Table 7. Functional Requirement – Submit Order

Functional Requirement	
기능	Submit order
설명	메뉴 확인 페이지에서 본인이 선택한 메뉴들을 확인하고 수정할 내용이 없어 주문을 넣으려면 주문하기 버튼을 클릭한다.
입력	주문하기 버튼
출력	대기 페이지(Final Page)
처리	장바구니 정보를 음식점 단말기로 전송, 주문정보 데이터베이스에 기록
조건	-

F. Final Page

Table 8. Functional Requirement – Final Page

Functional Requirement	
기능	Final Page
설명	손님이 주문을 완료했을 때, Final Page로 이동한다. Final Page에서는 손님의 주문이 잘 전달되었음을 표시한다.
입력	-
출력	Final Page 화면
처리	-
조건	-

G. Manage Orders

Table 9. Functional Requirement – Manage Order

Functional Requirement	
기능	Manage Orders
설명	식당 직원이 음식점 단말기에서 전달되어 온 주문서 정보를 주문 순서대로 볼 수 있게 한다. 하나의 웹페이지가 실시간으로 업데이트 되어 주문 정보들을 표시한다. 주문이 들어 왔을 때, 소리로 알람을 준다. 주문이 취소되거나 해당 음식을 조리, 서빙이 완료 되었을 때 주문서 정보를 웹 페이지에서 간단히 삭제할 수 있게 한다.
입력	주문서 정보
출력	대기 페이지
처리	장바구니 정보를 음식점 단말기로 전송, 주문정보 데이터베이스에 기록
조건	-

6.3. Non-functional Requirement

System 수준에서의 non-functional requirement를 기술한다. 크게 Product requirement, organizational requirement, external requirement가 있으며 각각 세부 항목으로 분류하여 기술한다. 기술 방식은 기본적으로 natural language를 사용하며, 필요시 mathematical notation을 추가하도록 한다.

A. Product Requirement

A.1. Performance Requirement

손님이 어떠한 단말기를 이용하던 상관 없이 일정한 시간 이내에 task를 마칠 수 있어야 한다.

Processed transactions between customer and restaurant : under 3 seconds

Event response time : under 0.5 seconds

Screen refresh time : under 3 seconds

A.2. Security Requirement

본 시스템 웹 페이지는 오직 매장 내 부착된 NFC를 통해서만 접근 가능해야 한다. 악의적인 주문을 방지하기 위해 종업원은 주문이 왔을 때 계산서를 테이블에 가져다 주고 메뉴 확인을 손님에게 부탁한다.

A.3. Efficiency Requirement

본 서비스는 식당 내의 사람들만 이용 가능하므로 서버에 접근 가능한 customer의 system 개수를 식당의 수용 가능 인원으로 한정한다. 이는 performance에 악영향을 미치지 않으면서 높은 효율의 시스템을 구성할 수 있게 한다.

A.4. Dependability Requirement

사용자의 정보, 즉 테이블 정보와 선택된 메뉴의 정보는 URL 쿼리를 통해 시스템간 정보를 주고 받는다. 따라서, URL 쿼리에 손상이 가지 않는 시스템을 구축해야 한다.

A.5. Usability Requirement

남녀노소 처음 시스템을 사용해본 사람이라도 편히 사용할 수 있어야 한다. 따라서 복잡한 기능을 최소화하고 UI/UX를 직관적으로 만들어야 한다.

B. Organization Requirement

B.1. Environmental Requirement

식당 테이블에 부착된 NFC 태그가 물리적으로 손상되어서는 안된다. 또한, NFC 내에 정보에 손상을 가할 수 있을 만한 자성을 띤 물체를 가까이 해서는 안된다. 또한, NFC 태그가 음식을 올려놓는 곳이 아닌 테이블의 가장자리에 있어야 하며, 내부 인테리어나 식사를 방해할 만큼의 크기나 모양이면 안된다. 마지막으로, 손님들 중 한명은 데이터 사용이 가능한 스마트폰을 지니고 있어야 한다.

B.2. Operational Requirement

음식점의 매니저, 웨이터, 셰프 등은 모두 이 시스템을 통해 주문내역을 확인하고 이를 수정할 수 있는 권한이 있어야 한다. 서빙이 완료된 주문내역을 삭제하는 것은 자주 일어나므로 간단한 동작(스 와이핑)을 통해 구현한다. 손님도 본인이 주문한 음식의 정보를 확인할 수 있어야 한다.

B.3. Development Requirement

사용자의 주문 내역 정보를 URL 쿼리에 저장하여 주문 과정을 거치는 동안 여러 정적 웹 페이지를 이동하는 동안 정보가 기억될 수 있도록 한다. 단말기의 종류에 영향을 덜 받는 웹 서비스를 채택하고, Django 프레임워크를 통해 하드웨어와 웹 서비스 간의 호환성을 높인다.

C. External Requirement

C.1. Legislative Requirement

축적되는 식당의 주문 내역 데이터를 빅데이터 분석에 활용하기 위해, 식당의 주문 내역에 접근하고 이를 이용할 수 있도록 식당 측의 동의를 얻어야 한다.

C.2. Safety/Security Requirement

본 시스템은 식당의 모든 음식 판매 및 주문 내역 정보를 데이터베이스로 관리하기 때문에, 정보 유출을 방지하기 위해 보안성을 높여야 한다. 또한 NFC에 인코딩한 URL이 외부에서 접근되지 않도록 한다.

6.4. Scenario

A. Setting Wizard Of Order

A.1 Normal Flow

오더의 마법사를 사용하기 위해서는 우선, 식당에 NFC 설치 및 메뉴 업로드가 필요하다. 클라이언트인 식당들은 오더의 마법사에 올릴 메뉴를 카테고리별로 나눠서 가격과 함께 제출한다. 제출한 데이터를 바탕으로 해당 식당 고유의 웹사이트를 제작한다.

이와는 별개로 각 테이블에 NFC 설치가 필요하다. 각 테이블에 부착된 NFC를 인코딩 할 때는 (?table_number=1)를 이용하여 테이블 고유의 넘버를 설정해준다.

A.2 Probable problems

웹페이지는 클라이언트 식당에서 제출하는 메뉴와 가격을 토대로 작성하여 식당에게 전달하지만, 추후에 변동이 생길 시 식당 측에서 손쉽게 웹페이지를 수정할 수 없다. 따라서 식당 측에서는 메뉴나 가격을 변동할 상황이 생길 시에는 웹페이지를 수정 할 수 있는 개발자를 고용할 필요가 있다.

A.3 System state on completion

Table 10. Menu Example

메뉴	카테고리	가격
김치볶음밥	밥류	5000
곤드레비빔밥	밥류	6000
김치찌개	찌개류	7000
된장찌개	찌개류	7000
콜라	음료	1000

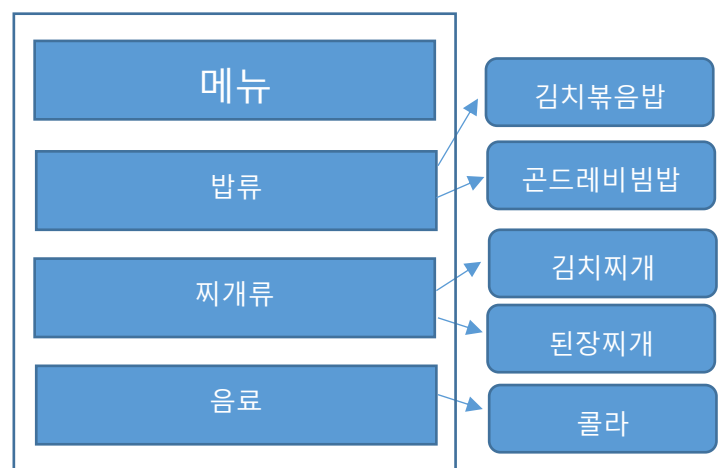


Diagram 6. Menu Example Diagram

B. Tagging NFC & loading webpage

B.1 Normal Flow

오더의 마법사가 설치된 식당을 방문하는 손님은 테이블에 부착된 NFC에 핸드폰을 가져다 대어 웹사이트에 접속한다. 해당 식당의 메뉴가 화면에 로드된다.

B.2 Probable problems

손님이 핸드폰을 보유하고 있지 않은 경우 오더의 마법사를 이용할 수 없다. 이런 경우는 직접 포스에 입력하는 방식으로 주문을 받아야 한다. 이전에 방문했던 손님이나 URL주소를 아는 타인이 매장에 방문하지 않은 채 웹페이지에 접속이 가능할 수 있다. 즉 보안적으로 문제가 생길 수 있다. 이러한 경우는 소프트웨어적으로는 2가지 방식으로 해결 할 수 있다. 첫째, URL주소를 주기적으로 변경한다. 둘째, 주문창에 비밀번호를 입력하는 란을 만들고, 식당 잘 보이는 곳에 비밀번호를 공지해 놓는다. 오더의 마법사에서는 위와 같은 번거로운 작업을 추가하기 보다는 주문리스트창에 새로운 주문이 뜨면 알바생이 직접 손님에게 계산서를 가져다 주면서 육안으로 테이블에 손님이 있는지 확인을 하는 방식을 채택하기로 했다.

B.3 System state on completion

해당 식당의 메뉴가 핸드폰 화면에 로드된다.



Figure 3. Main Page

C. Collect and confirm the menu

C.1 Normal Flow

손님들은 메뉴를 고르고 수량을 정해서 오더를 전송한다. 이는 다음과 같은 단계들로 이루어진다.

첫 화면에서 카테고리를 고른다. (ex. 밥류, 찌개류, 면류, 음료)

해당 카테고리에서 구체적인 메뉴를 고른다.

선택한 메뉴의 수량을 설정한다.

주문버튼을 눌러 주문확인 화면으로 넘어간다.

주문이 올바르게 되었는지 확인을 하고 확인 버튼을 눌러 최종적으로 오더를 전송한다.

C.2 Probable problems

이전에 방문했던 손님이나 URL주소를 아는 타인이 매장에 방문하지 않은 채 웹페이지에 접속이 가능할 수 있다. 즉 보안적으로 문제가 생길 수 있다. 이러한 경우는 소프트웨어적으로는 2가지 방식으로 해결 할 수 있다. 첫째, URL주소를 주기적으로 변경한다. 둘째, 주문창에 비밀번호를 입력하는 란을 만들고, 식당 잘 보이는 곳에 비밀번호를 공지해 놓는다. 오더의 마법사에서는 위와 같은 번거로운 작업을 추가하기 보다는 주문리스트창에 새로운 주문이 뜨면 알바생이 직접 육안으로 테이블에 손님이 존재하는지 확인을 한 뒤 컨펌을 하는 방식을 채택하기로 했다.

C.3 System state on completion

메뉴와 수량을 선택한 뒤 주문 버튼을 누르면 주문확인 화면을 거친 뒤, 주문완료 페이지가 핸드폰 화면에 로드된다.



Figure 4. Menu Page



Figure 5. Final Page

D. Inform order

D.1 Normal Flow

서버에 오더가 전송되면 서버에서 주문을 주문 큐에 입력한다. 주문리스트가 업데이트되면, 조리담당자들은 주문리스트를 확인하고 조리를 시작한다. 따라서 주문리스트는 먼저 온 주문부터 순차적으로 리스트를 보여주어야 하며, 서브메뉴 등과 함께 오더를 자세하게 보여줄 필요가 있다. 한편, 다른 알바생은 주문리스트를 보고 해당 오더의 테이블 넘버를 참조하여 POS에 주문을 입력한다. 조리자가 조리를 끝내면 음식을 서빙 담당자에게 전달하고, 리스트에서 해당 주문을 삭제한다.

D.2 Probable problems

이미 오더가 들어간 상황에서 손님이 주문을 수정하고자 하는 상황이 생길 수 있다. 이러한 상황에서는 식당 내에서

D.3 System state on completion

주문리스트 화면에 출력될 내용은 다음과 같다.

테이블 넘버

오더가 들어온 시간

오더 상세사항

Table 11. Example – Order History

Table number	Time	Order
3	16:34	김치볶음밥(치즈추가), 콜라
10	16:40	김치찌개, 콜라
7	17:01	곤드레비빔밥, 김치볶음밥, 콜라(2)
1	17:12	곤드레비빔밥(밥추가)

E. Record order

E.1 Normal Flow

오더가 최종적으로 완료되면 조리자 측의 주문리스트 외에도 주문기록 데이터베이스에 오더가 입력된다. 기록된 주문내역은 추후에 통계적 분석을 위해 사용된다.

E.2 Probable problems

주문내역 데이터를 기반으로 통계적 분석을 진행할 또다른 소프트웨어가 필요하다. 또한, 허위주문이나 주문변동사항이 생길 시 무의미한 데이터가 입력될 가능성이 있다.

E.3 System state on completion

주문기록 데이터베이스에 입력되는 데이터는 주문리스트에 출력되는 내용과 일치한다.

7. System Models

7.1. Objective

System model에서는 시스템을 구성하는 서브 시스템, 그리고 컴포넌트들의 상호작용과 그 관계를 설명한다. Object model, data-flow model 그리고 semantic data model 등을 활용해 작성한다.

7.2. Context Models

A. Context Model

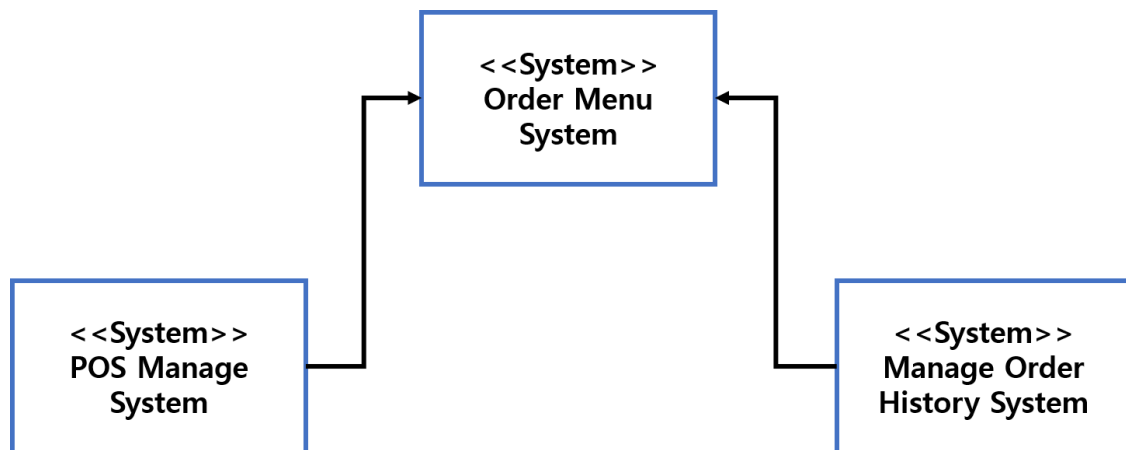


Diagram 7. Context Diagram

B. Process Diagram

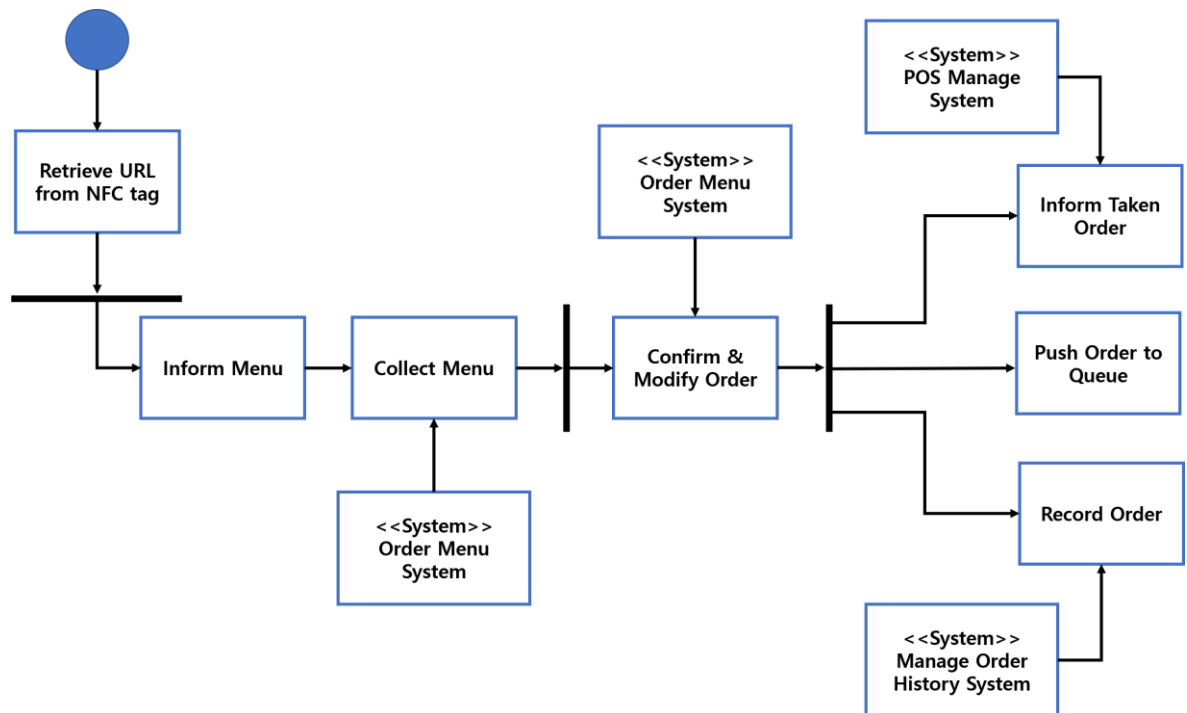


Diagram 8. Process Diagram

7.3. Interaction Models

A. Use Case Models

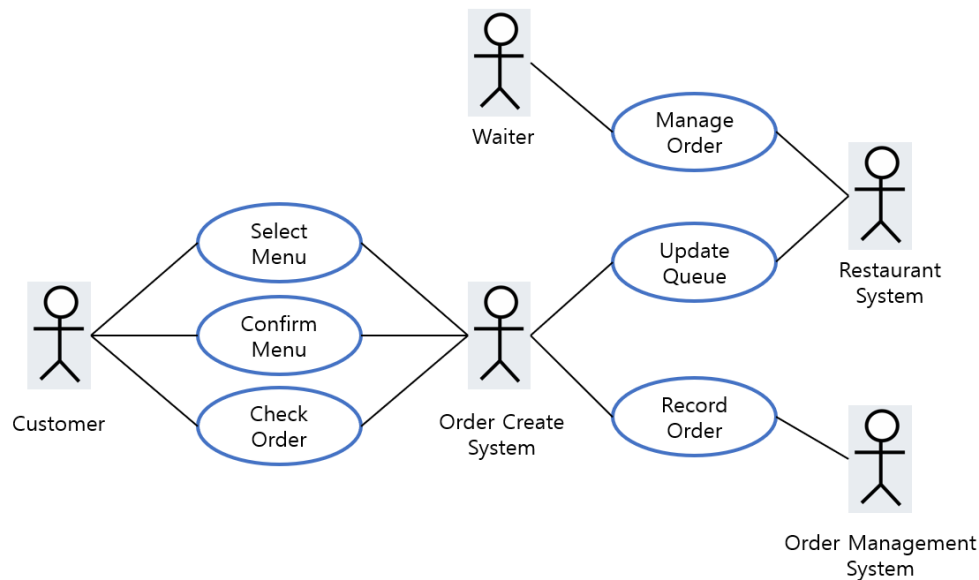


Diagram 9. Use Case Diagram

B. Tabular description for each use case

B.1. Select Menu

Table 12. Use Case – Select Menu

Select Menu	
USE CASE	Select Menu
ACTOR	Customer, Order Create System
DESCRIPTION	<ul style="list-style-type: none"> -Customers pick the items they want to order from the menu and place the foods in their order list. -The information of the food to be entered into the menu is retrieved from the order create system by accessing the food database.
STIMULUS	<ul style="list-style-type: none"> -Customer requests a restaurant menu web page for order creation via request transfer to order create system. -When customers complete their order, they will again request

	an order confirmation page to confirm the menu on the order create system.
RESPONSE	<ul style="list-style-type: none"> -The order create system invokes the menu information from the food database and creates a restaurant menu web page and sends it when a customer request arrives. -The order information is included in the URL query so that the customer can confirm his / her menu information in the menu confirmation.
COMMENTS	-The customer accessing the order create system must be a user accessing via NFC tag.

B.2. Confirm Menu

Table 13. Use Case – Confirm Menu

Confirm Menu	
USE CASE	Confirm Menu
ACTOR	Customer, Order Create System
DESCRIPTION	<ul style="list-style-type: none"> -Customer checks the order list which is created from the select menu and returns to the select menu stage to complete the order, or to modify it. -Confirm menu step also provides a function to easily adjust the quantity of the menu or to delete a specific menu from the order book.
STIMULUS	<ul style="list-style-type: none"> -When Customer completes the order list from the select menu stage, the order create system requests the web page for confirm menu stage via request. -If Customer confirms the order, click the Order Complete button to send a request to finalize the order. -If you want to modify the menu, send a request to adjust it on the confirm menu web page or return to the select menu. -The final order information is sent to the restaurant system and order management system, which records them in a database in each system.
RESPONSE	<ul style="list-style-type: none"> -Order create system sends a web page that shows the order information according to the customer's confirmation request. -If Customer requests a menu adjustment, adjust the order

	<p>information at this stage or resend the select menu web page.</p> <p>-When Customer confirms the order, it sends the final order information to the restaurant system and the order management system. And notifies the customer that the order has been completed.</p> <p>-The order create system sends the final order information to the restaurant system and the order management system.</p>
COMMENTS	<p>-When the Customer returns to the select menu again, the URL must contain order information in the form of a query. If there is an order adjustment in the confirm menu, it must be reflected in the query.</p>

B.3. Check Order

Table 14. Use Case -Check Order

Check Order	
USE CASE	Check Order
ACTOR	Customer, Order Create System
DESCRIPTION	<p>-When Customer confirms the order, the order create system displays a web page informing that the order has been completed.</p> <p>-If Customer wants to place an additional order, you can proceed with the select menu again.</p>
STIMULUS	<p>-When the customer confirms the order in the Confirm menu, the customer sends a request to the order create system to request the check order web page.</p> <p>-If you want to place additional orders on the final page, send a request to the order create system requesting a select menu web page.</p>
RESPONSE	<p>-Order create system receives a customer's check order request and generates a web page for the system and sends it.</p> <p>-According to the request for additional order, the order create system again generates a select menu web page and sends it to the guest.</p>
COMMENTS	

B.4. Manage Order

Table 15. Use Case – Manage Order

Manage Order	
USE CASE	Manage Order
ACTOR	Waiter, Restaurant System
DESCRIPTION	<p>-Waiter receives order information from the restaurant system to be prepared at the restaurant. The restaurant system lists the orders in the order they were requested.</p> <p>-If the food is cooked and served or the order is canceled, the waiter will ask the restaurant system to remove the order information.</p>
STIMULUS	<p>-The restaurant system receives the order information from the database and arranges it in the order of the list. Based on this, the manage order web page is created and transmitted to the terminal used by the restaurant.</p> <p>-Waiter sends a request to the restaurant system requesting the removal of the completed or canceled order from the order list.</p>
RESPONSE	-The restaurant system updates the list as the waiter corrects the order form, generates the web page again, and sends it to the terminal.
COMMENTS	-When developing the system, we review the order information in real time from the inside of the store terminal and check whether it is possible to build a system that can output and check it.

B.5. Update Queue

Table 16. Use Case – Update Queue

Update Queue	
USE CASE	Update Queue
ACTOR	Order Create System, Restaurant System
DESCRIPTION	-When the customer confirms the order, the order create system sends the final order information to the restaurant system. This information is used as a list of foods the waiter will serve.

	-The restaurant system processes the order information received from the order create system as a waiting queue and configures the web page so that the store terminal can process it.
STIMULUS	-When the user makes a final order from the confirm menu, the order create system sends the order information to the restaurant system.
RESPONSE	-The restaurant system stores the order information in the system's cooking order database.
COMMENTS	-The order information must include a time attribute because it must be queued sequentially.

B.6. Record Order

Table 17. Use Case -Record Order

Record Order	
USE CASE	Record Order
ACTOR	Order Create System, Order Management System
DESCRIPTION	-When the customer confirms the order, the order create system sends the final order information to the order management system. The information is cumulative and can be used for marketing analysis of the restaurant in future evolution.
STIMULUS	-If the customer makes a final order in the confirm menu, the order create system sends the order information to the order management system.
RESPONSE	-The order management system stores the order information in the system's cooking order database.
COMMENTS	-It is a database that is constantly accumulated and contains restaurant information, so you need to be careful about security so that internal data is not leaked.

C. Sequence Diagram

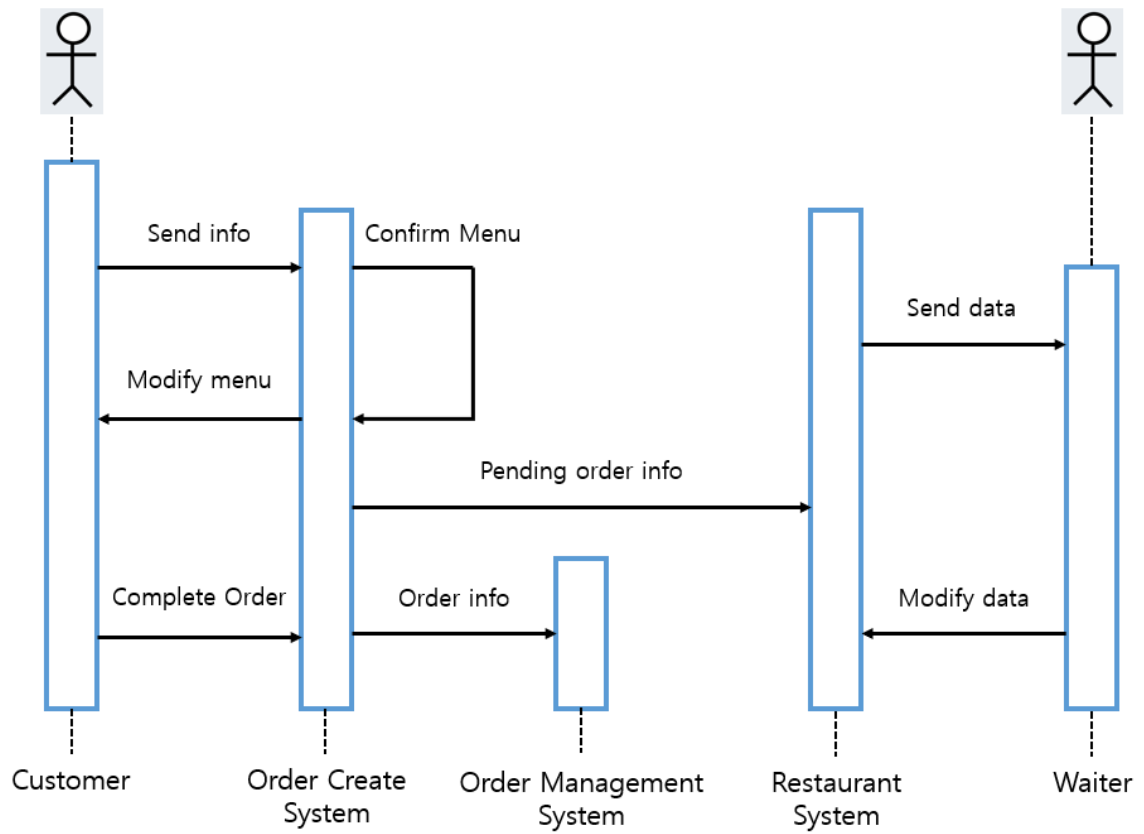


Diagram 10. Sequence Diagram

7.4. Structural Models

A. Class Diagram

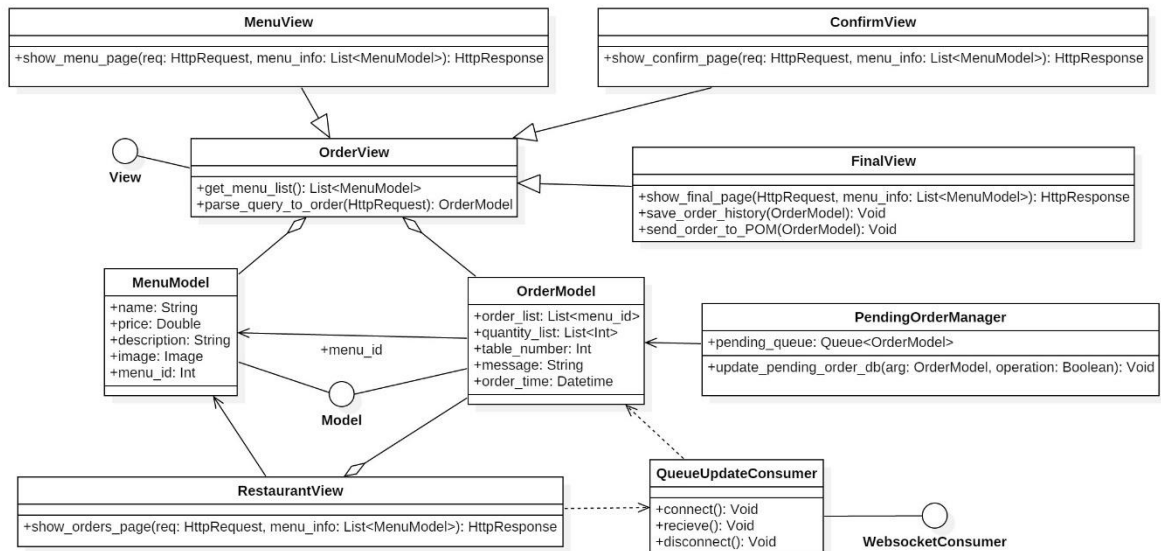


Diagram 11. Class Diagram

7.5. Behavioral Models

A. Event-Driven Modelling

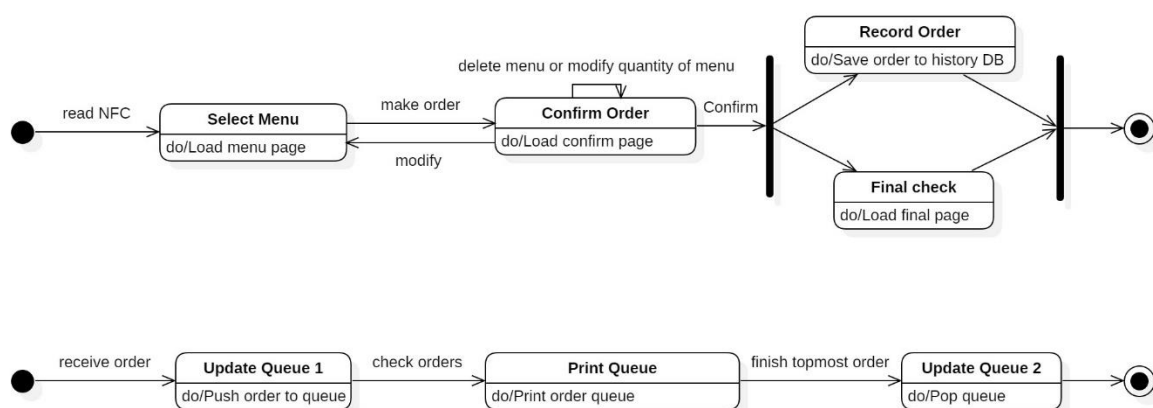


Diagram 12. Event-Driven Model : State Diagram

8. System Evolution

8.1. Objective

System evolution에서는 시스템을 위한 주요 가정들에 대해 설명하고, 발생할 수 있는 변화를 예측한다. 이는 하드웨어의 발전, 사용자의 요구사항 변화 등에 기반한다. 나아가 어떻게 이러한 변화사항을 어떻게 반영할 수 있는지를 살펴본다. 이는 시스템 설계자가 추후에 시스템의 변경을 피할 수 있도록 해주고 변화에 적응가능한 시스템을 개발할 수 있도록 도와준다.

8.2. Limitation and Assumption

아직까지 대부분의 소비자는 주문 자동화 시스템에 친숙하지 않다. 간혹, 키오스크를 쓰는 매장은 종종 보이고 그 수가 늘어가고 있지만 한편으로는 키오스크로 인한 불편함 또한 존재한다. 2. Introduction에서 설명했듯 본 시스템의 필요성은 충분하지만 한계점 또한 존재한다. 우선, 본 시스템은 결제를 먼저 해야 하는 식당에서는 도입할 수가 없다. 타겟으로 하는 식당들은 추가적인 주문이 계속해서 들어오는 고깃집, 술집 혹은 들어오는 주문을 기존 직원의 수로 감당하지 못할 만큼 직원 수는 곳을 타겟으로 한다. 즉, 일하는 직원의 일손을 덜어주는 것이 시스템의 목적이다. 하지만, 클라이언트 식당에 따라서 선결제를 원하는 매장이 있을 수 있다. 가령, 후결제 방식에서는 음식을 받고나서 결제를 하지 않은 채 나가는 손님이 생길 우려가 있기 때문이다. 만일 선결제 방식을 도입한다면, 서빙과 조리 이외에도 결제까지 담당해야하는 직원들의 일손을 더욱 덜어줄 수 있을 것이다.

한편, 본 시스템은 주문 내역 데이터를 수집할 수 있도록 고안되어 있다. 손님이 오더를 최종적으로 완료하면 조리자 측의 주문리스트 외에도 주문기록 데이터베이스에 오더가 입력된다. 기록된 주문내역은 추후에 통계적 분석을 위해 사용된다. 이와 같이 기록되는 주문 내역 데이터는 Pending Order Database에 저장되지만 이를 분석하고 활용할 수 있는 소프트웨어는 제공되지 않는다.

8.3. Evolution of Review platform

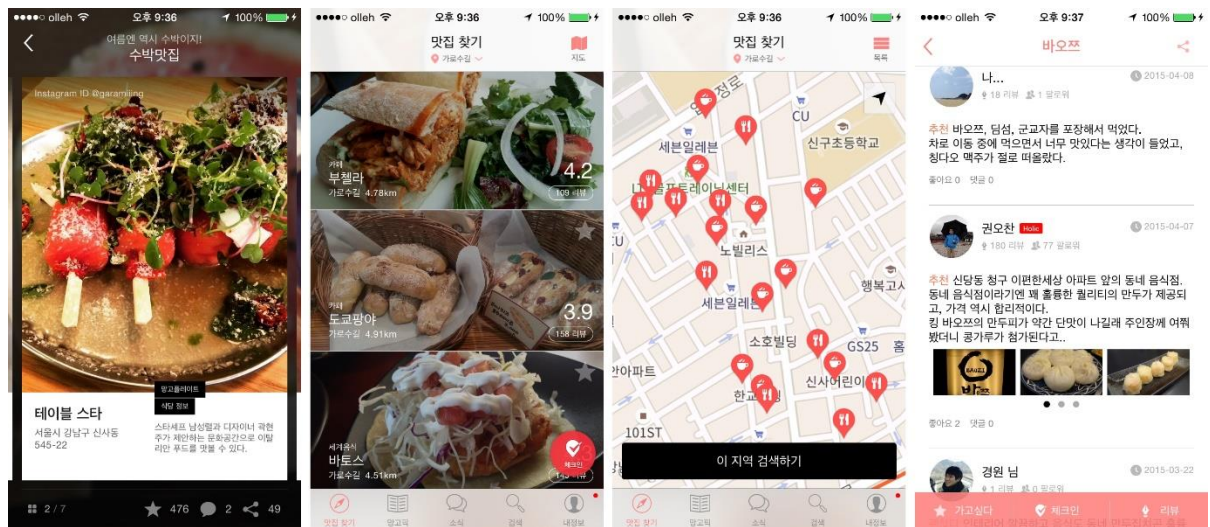


Figure 6. Display of MangoPlate application

오늘날에는 개인의 경험이나 주변사람들의 입소문보다는 SNS를 통해 필요한 정보를 얻는 경향이 더욱 짙어지고 있다. 고객들은 밥 한끼를 먹더라도 보다 더 저렴하고 맛있는 곳을 방문하길 원한다. 이러한 욕구는 위의 사진에서 볼 수 있는 망고플레이트와 같은 식당 리뷰 어플리케이션이 활성화되고 있다는 점에서 드러난다. 이와 같이 리뷰 플랫폼에 대한 needs와 user가 계속해서 증가한다면 본 시스템에서도 리뷰 플랫폼을 제공할 필요성이 충분히 생긴다.

8.4. Evolution of Bigdata Marketing

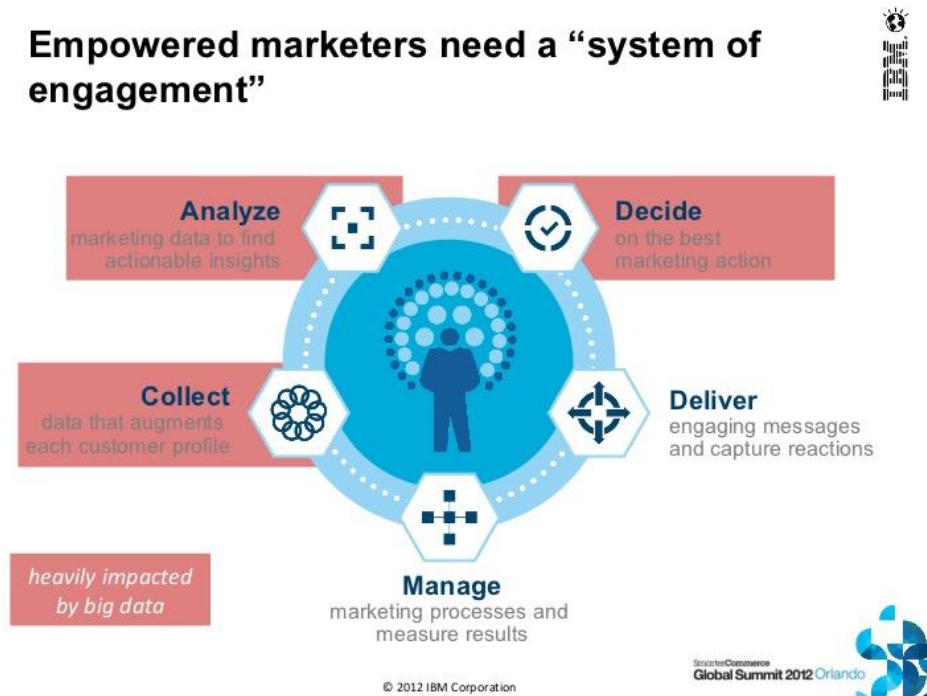


Figure 7. Need of Big data on Marketing

빅데이터 마케팅이란, 빅 데이터를 통해 고객의 소비패턴과 선호도, 정보 등을 분석하여, 구매할 가능성이 높은 고객에게 맞춤형 혜택을 제공하는 것을 말한다. 현존하는 빅데이터 마케팅으로는 네이버 뮤직 라디오의 음악 추천 시스템, 미처 생각하지 못했지만 꼭 필요한 제품까지 추천해주는 아마존의 서비스, 소비자가 좋아하는 브랜드의 쿠폰만 보내주는 백화점 서비스 등을 예로 들 수 있다. 기업에서는 이러한 빅데이터 마케팅을 통해 마케팅 반응률을 크게 높일 수 있고 나아가 탄탄한 비즈니스 모델을 형성할 수 있다. 본 시스템에서는 Pending Order Database 및 리뷰 플랫폼을 이용하여 빅데이터를 수집할 수 있을 것으로 보인다. 나아가 수집한 데이터를 분석하는 분석시스템을 개발한다면 이와 같은 빅데이터 마케팅의 활용에 도움이 될 수 있을 것으로 보인다.

8.5. Evolutions of User Requirement

현재 고안된 오더의 마법사 시스템은 결제시스템을 지원하고 있지 않다. 하지만, 결제 모듈 및 선결제 방식을 도입하길 원하는 사용자가 존재할 수 있다. 따라서 본 시스템에서 최종 오더를 완료한 뒤 결제를 할 수 있도록 개발을 진행한다면, 서빙과 조리 이외에도 결제까지 담

당해야하는 직원들의 일손을 더욱 덜어줄 수 있을 것이며 이에 따라 user requirement를 더욱 충족시킬 수 있을 것으로 보인다.

또한, 현재 고안된 오더의 마법사 시스템은 식당 주인이 직접 주문내역 데이터를 활용하여 데이터 분석을 진행 할 수 있는 소프트웨어를 제공하고 있지 않다. 하지만, 소규모 혹은 개인 식당에서 주문과 관련된 전산 데이터를 관리하고 이를 활용하는 일은 쉽지 않다. 가게의 음식 판매량 등을 시각화 해주고, 가게 주인이 이를 마케팅에 활용할 수 있는 솔루션을 제안해준다면 데이터분석에 관련하여 예상되는 user requirement를 충족시킬 수 있을 것으로 보인다.

8.6. Evolutions of Environment

본 시스템은 손님이 매장을 방문하여 겪는 주문시스템을 간편화하기 위한 방안에 초점을 두고 있다. 이와는 별개로 손님들이 등록된 식당들의 리뷰를 검색하고 참고할 수 있는 리뷰 플랫폼을 구축한다면 손님들의 user requirement를 더욱 충족시킬 수 있을 것으로 보인다. 우선, 리뷰 시스템을 활용해 등록된 식당들의 리뷰를 볼 수 있는 플랫폼을 구축한다. 그리고 결제를 하고 리뷰를 할 경우, 소비자에게 포인트 혹은 플랫폼 내에서 활용할 수 있는 이점을 제공하여 리뷰를 유도한다. 이는 실제로 매장에 방문한 사람이 쓸 수 있는 리뷰이므로 이는 정보의 신뢰성이 보장된다. 따라서, 다른 사용자는 정확한 정보를 가지고 자신이 가고 싶은 식당을 찾을 수 있다.

다음으로 매장 주인의 user requirement로 빅데이터 마케팅을 위한 데이터 분석이 요구될 수 있다. 본 시스템에서는 Pending Order Database 및 리뷰 플랫폼을 이용하여 빅데이터를 수집할 수 있을 것으로 보인다. 나아가 수집한 데이터를 분석하는 분석시스템을 개발한다면 빅데이터 마케팅의 활용에 도움이 될 수 있을 것으로 보인다. 결과로 제공되는 데이터는 매장에서 주요고객 및 잠재적 고객들을 관리하는 도구로 이용될 수 있을 것이다.

9. Appendices

9.1. Objective

Appendices에서는 개발되는 시스템에 대한 구체적이고 자세한 정보를 제공한다.

9.2. Database Requirements

이 절에서는 구축하고자 하는 오더의 마법사 Database의 요구사항을 서술한다. 다음에서 설명하는 내용들은 해당되는 요구사항에 알맞는 형태로 데이터베이스 테이블에 저장되어야 한다. 각 요구사항에 대해서는 Attribute 는 *이탤릭체*, Entity type은 **굵은 글씨체**, Relationship 은 **굵은 이탤릭체**, 제약조건(Constraints)는 밑줄로 표시하였다.

오더의 마법사에는 세 가지 Entity가 필요하다. 첫째, Menu의 목록을 저장할 테이블이 필요하다. 이는 **FoodDB**로 불린다. **FoodDB**는 필수적으로 name, price, menu_id를 attribute로 가지며, 추가적으로 *description, image*를 갖는다. 다음으로, OrderDB 는 들어온 주문을 저장한다. **OrderDB**는 필수적으로 테이블번호와 menu_id, name, quantity를 attribute로 가지며, 추가적으로 *주문비고사항을 저장하는 message, 주문이 들어온 시간을 저장하는 order_time*을 attribute으로 갖는다. **PendingOrderDB**는 **OrderDB**와 동일한 attribute를 갖는다.

9.3. Database details

FoodDB는 OrderView기능을 위한 MenuModel에서 참조되어 사용된다. FoodDB는 초기 매장 등록시에 데이터를 입력하며, 추가적으로 변동사항이 생길 때 수정될 수 있다. OrderDB는 주문이 최종적으로 완료되고나서 Order를 저장한다. 저장된 주문내역은 추후에 데이터분석에 쓰일 수 있다. PendingOrderDB는 현재 처리되지 못한 주문목록을 저장한다. 테이블의 형식과 저장되는 attribute들의 형태는 OrderDB와 동일하지만, 처리된 오더는 주문리스트에서 삭제되는 동시에 PendingOrderDB에서도 delete된다는 점에서 상이하다.

9.4. User-System Requirements

주문리스트를 출력할 화면으로는 복잡한 프로세스를 요구하지 않으므로 고사양의 하드웨어가 필요하지 않다. 고객이 사용하는 모바일 디바이스 사양으로는 NFC기능이 요구되며, 웹페이지를 로드하기 위해서는 http를 읽을 수 있는 스마트폰 사양이 요구된다.

9.5. Development process

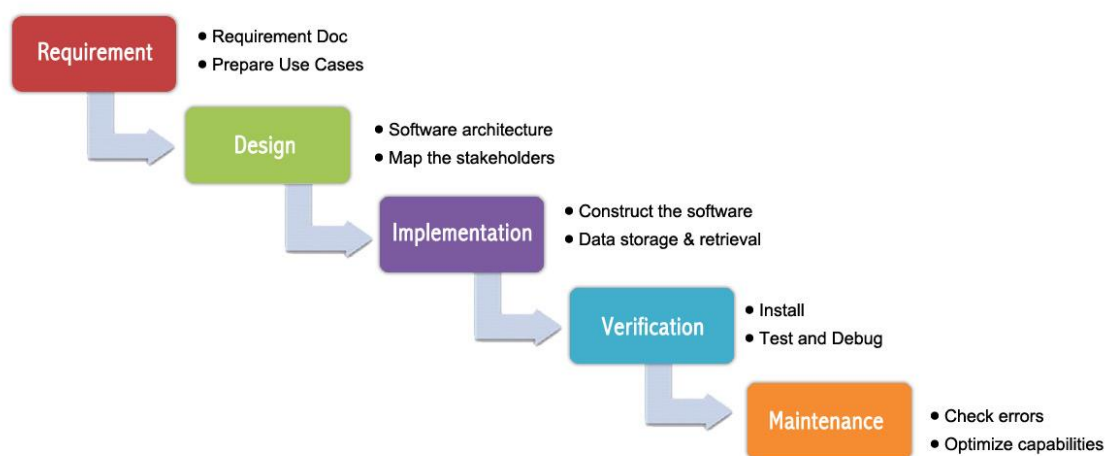


Figure 8. Waterfall Model Diagram

본 시스템은 requirement들이 명확하게 제시되어 있고, 앞으로의 변화 또한 결제 모듈을 추가하는 등 어느정도 예상이 가능하다. 또한 시스템의 특성상 유지, 보수가 중요한 부분이기 때문에 이를 좀 더 용이하게 하기 위하여 문서가 자세히 작성할 필요가 있으며, parallel development가 가능하다. 따라서 본 프로젝트는 Waterfall process를 따라서 개발할 예정이다.

Waterfall model은 계획 기반 모델로 소프트웨어 프로세스의 각 단계를 나누고 구별한 모델로, 모든 각각의 단계에서는 산출물이 명확하고, 또한 문서가 자세하고 정확히 작성된다. 그리고 병렬적 개발이 가능하기 때문에 일반적으로 많이 쓰이는 모델이며, 특히 대규모 프로젝트에서 쓰인다.

10. Index

10.1. Objective

Index에서는 본 문서에서 사용된 용어와, 다이어그램, 기능에 대한 인덱스를 나타낸다.

10.2. Table Index

Table 1 Term definition.....	13
Table 2 Acronyms and Abbreviations.....	14
Table 3. Functional Requirement – Set NFC Tag.....	23
Table 4. Functional Requirement – Load Webpage.....	24
Table 5. Functional Requirement – Select Menu.....	25
Table 6. Functional Requirement – Confirm Order	25
Table 7. Functional Requirement – Submit Order	26
Table 8. Functional Requirement – Final Page	26
Table 9. Functional Requirement – Manage Order	27
Table 10. Menu Example	30
Table 11. Example – Order History	34

10.3. Figure Index

Figure 1. Kiosk Market Outlook	9
Figure 2. Tablepad example	10
Figure 3. Main Page.....	31
Figure 4. Menu Page	33

Figure 5. Final Page	33
Figure 6. Display of MangoPlate application.....	46
Figure 7. Need of Big data on Marketing	47
Figure 8. Waterfall Model Diagram	50

10.4. Diagram Index

Diagram 1. Entire System.....	11
Diagram 2. Architectural Diagram	19
Diagram 3. Architectural Diagram – Order Create System	20
Diagram 4. Architectural Diagram – Order Management System.....	21
Diagram 5. Architectural Diagram – Restaurant System	22
Diagram 6. Menu Example Diagram.....	30
Diagram 7. Context Diagram.....	36
Diagram 8. Process Diagram	37
Diagram 9. Use Case Diagram	38
Diagram 10. Sequence Diagram	43
Diagram 11. Class Diagram.....	44
Diagram 12. Event-Driven Model : State Diagram	44

11. Reference

인용 자료

Figure 1. Researchbbc. (2016). "Kiosk Market Outlook." MarketsandMarkets.

Figure 1. 김용균. (2017년 4월 5일). 무인화 추세를 앞당기는 키오스크. "주간기술동향", 페이지: 18.

Figure 2.

<https://m.blog.naver.com/PostView.nhn?blogId=tpgml1004_&logNo=220837523383&categoryNo=0&proxyReferer=&proxyReferer=http%3A%2F%2Fwww.google.com%2Furl%3Fsa%3Dt%26rct%3Dj%26q%3D%26esrc%3Ds%26source%3Dweb%26cd%3D13%26ved%3D0ahUKEwjG8Pj9nfHaAhVljFQKHZH8Cc0QFghPMAw%26url%3Dhttp%253A%252F%252Fm.blog.naver.com%252FPostView.nhn%253FblogId%253Dtpgml1004_%2526logNo%253D220837523383%2526categoryNo%253D0%2526proxyReferer%253D%26usg%3DAOvVaw1IaIndJCmL7A1kfk_6u8rX>

Figure 7. <<https://www.slideshare.net/graemeknows/big-data-bigger-campaigns-using-ibms-unica-and-netezza-platforms-to-increase-marketing-roi>>

Figure 6. <<http://www.newstomato.com/ReadNews.aspx?no=580580>>

네이버 지식백과, 빅데이터 마케팅,

<<https://terms.naver.com/entry.nhn?docId=2070422&cid=55570&categoryId=55570>>

위키백과, 빅데이터 마케팅,

<https://ko.wikipedia.org/wiki/%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0_%EB%A7%88%EC%BC%80%ED%8C%85>