

푸드 박스 시스템

요구사항 정의서

배달 안심스티커에도 음식 빼먹는 배달거지, "너무 찝찝해서 못 먹겠어요"

유해지 기자 | 승인 2020.02.15 00:32 | 댓글

재보하기 | 반종료

(톱스타뉴스 유해지 기자) 배달 안심 스티커가 붙어 있어도 음식을 빼먹는 일이 발생하고 있다.

지난 14일 한 온라인 커뮤니티에는 '배달'용인스티커도 소중 없는 배달종이라는 제목으로 글이 게재됐다.

해당 글에는 배달 업체 후기를 작성한 한 네티즌의 사진이 담겨 있다. 사진의 작성자는 "후장 스티커가 붙어 있길래 전황하니 열기 때문이라고 하시는데 아무래도 아닌 것 같다"고 말했다.



온라인 커뮤니티

그러면서 "먹 한 조각 만큼 자라도 비교 소스 자극도 거기만 딱 있었던 자극도 있고 진짜 너무 찝찝해서 못 먹겠네요. 자주 시켜 먹었는데 이제 믿거네요(믿고 거른다)"라고 덧붙였다.

이를 접한 네티즌들은 "열기에 찢어질 뿐인은 뭐하러 하나", "붙겨 있다는 의미 전달이 잘못된 거 같은데", "배달할 땐 찢어있음 수취거부 해야죠", "붙 붙이고 붙었다 말이 되는 소리인가요", "붙이 붙여 있으면 판매자가 양심상 교환해주고 그 비용을 배달 업체에 돌려야 정상이죠", "붙이 열기 때문에 저렇게 찢긴다고?" 등의 반응을 전했다.

한편 일부 배달원들은 배달 음식을 들며 빼먹는 '팁'을 온라인 커뮤니티에 게재하며 논란을 사기도 했다. 자신을 배달원이라고 밝힌 A씨는 안전-위험 단계 별로 빼먹기 쉬운 음식을 소개하며 그 방법을 그대로 전수하기도 했다.

이를 방지하고자 안심 스티커가 대책 방법으로 등장했지만 그럼에도 배달 사고는 끊이지 않고 있는 상황이다.

유해지 기자 reporter@topstarnews.co.kr

기자의 보도 내용에 비문을 수정하며 인용문을 보강합니다. 그러나 객관적인 사실이나 법칙에 대해서는 함의 없는 정사 자료를 주된자료로 유지합니다. 또한 톱스타뉴스의 모든 콘텐츠(영상, 기사, 사진)는 저작권법의 보호를 받습니다. 무단 전재, 복사, 배포, 유통 등 영상 제작을 금합니다. 발견 즉시 민형사상 제소합니다.

Tag: #배달 #안심스티커

작성자: 안성현

설계자: 안성현, 이동준, 밥킨 다니일, 이강주

확인자: 안성현, 이동준, 밥킨 다니일, 이강주

목차

- 1. 개요 ...3
- 2. 개발 SW의 주요 기능 ...3
- 3. 개발 SW의 범위 ...3
- 4. 개발 SW의 조건 ...4
- 5. 요구사항 목록 ...5
- 6. 요구사항 명세 ...6
 - 6.1. ID ...6
 - 6.2. Configuration ...7
 - 6.3. Communication8
 - 6.4. Device ...9
 - 6.5. Store ...10
 - 6.6. Consumer ...11
 - 6.7. BackEnd ...12

1. 개요

본 문서는 푸드 박스 시스템을 개발하기 위한 사용자 요구사항을 정의한 문서이다. 본 문서의 목적은 사용자 요구 사항을 정의함으로써 개발 과정에서 발생하는 개발자와 설계자, 사용자 간의 대화 불일치를 방지하고 목표 시스템의 목적과 기능을 명확히 한다.

2. 개발 SW의 주요 기능

- 가게는 ID, 가게명, 비밀번호 등의 정보로 회원 가입을 할 수 있다.
- 가게는 로그인을 할 수 있다.
- 가게는 로그아웃을 할 수 있다.
- 소비자는 ID, 비밀번호 등의 정보로 회원 가입을 할 수 있다.
- 소비자는 로그인을 할 수 있다.
- 소비자는 로그아웃을 할 수 있다.
- 소비자는 가게를 선택하여 배달 주문을 요청할 수 있다.
- 가게는 요청된 배달 주문 목록을 조회할 수 있다.
- 가게는 요청된 배달 주문 정보를 조회할 수 있다.
- 가게는 요청된 배달 주문에 대해 사용할 기기를 QR 코드 스캔으로 할당할 수 있다.
- 가게는 요청된 배달 주문에 할당된 기기에 대해 할당을 해제할 수 있다.
- 가게는 '배달 시작' 버튼을 눌러 주문을 요청한 소비자에게 배달 시작 알림을 전송할 수 있다.
- 소비자는 '잠금 해제' 버튼을 눌러 기기의 잠금을 해제하고 가게에게 배달 완료 알림을 전송할 수 있다.
- 가게는 기기의 GPS 정보를 조회할 수 있다.
- 소비자는 주문에 할당된 기기의 GPS 좌표를 조회할 수 있다.
- 가게는 기기의 상태(배달 준비, 배달 중, 배달 완료) 정보를 조회할 수 있다.
- 가게는 배달 완료 상태의 기기의 주문 할당을 해제할 수 있다.

3. 개발 SW의 범위

- 개발 SW는 기존의 배달 앱 시스템에 추가될 기능에 대한 SW이다.
- 가게는 고정된 메뉴를 가지고 있으며 메뉴에 대한 CRUD는 개발하지 않는다.
- 소비자는 가게를 선택하여 주문하며 가게의 메뉴를 선택할 수 없다.
- 개발 SW 중 프론트엔드 사이드는 기기 클라이언트 SW, 가게 클라이언트 SW, 소비자 클라이언트 SW로 구성된다.
- 개발 SW 중 백엔드 사이드는 IoT 허브 SW 및 DB, 앱 서버 SW 및 DB로 구성된다.
- IoT 허브 SW는 IoT DB와 통신한다.

- 앱 서버 SW는 앱 DB와 통신한다.
- 기기 클라이언트 SW는 오직 IoT 허브와 통신한다.
- 가게 클라이언트 SW와 소비자 클라이언트 SW는 오직 앱 서버 SW와 통신한다.
- 앱 서버 SW는 IoT 허브와 통신한다.
- 기기 클라이언트 SW는 Python과 Bash Shell Script로 개발하여 라즈베리파이 OS 상에서 작동한다.
- 서버 SW는 Spring Boot Framework로 개발하여 네트워크 클라우드 상에서 작동한다.
- 백엔드 SW의 DBMS로 MariaDB를 이용한다.
- 가게 및 소비자 클라이언트 SW는 React Native Framework로 개발하여 Android OS, IOS 상에서 작동한다.

4. 개발 SW의 조건

- 가게, 소비자는 로그인 세션을 유지할 수 있다.
- 기기는 GPIO를 제어할 수 있어야 한다.
- 기기는 고유 식별번호에 대한 로컬 파일을 가져야 한다.
- 기기, 가게, 소비자 앱은 서버 IP/Port에 대한 암호화된 로컬 파일을 가져야 한다.
- 기기, 가게, 소비자 앱은 각 시스템 모듈의 로컬 파일에 대한 읽기 권한을 가져야 한다.
- 기기, 가게, 소비자 앱은 암호화된 서버 IP/Port에 대한 복호화를 할 수 있어야 한다.
- 기기는 주기적으로 서버 측으로 배터리 잔량 정보를 전송한다.
- 가게는 배터리 잔량이 25% 이하인 기기에 주문을 할당할 수 없다.
- 기기에 주문이 할당되면 주문 할당 안내 LED를 점등해야 한다.
- 기기의 주문 할당 및 해제, 잠금장치 활성화 및 비활성화 시 스피커로 안내음을 출력해야 한다.
- 기기, 가게, 소비자는 네트워크에 연결되어 있어야 한다.
- 기기, 가게, 소비자는 HTTP와 웹소켓을 이용하여 백엔드 서버를 통해 통신한다.
- 기기, 소비자는 GPS 통신이 가능해야 한다.
- 기기는 잠금장치의 활성화 여부를 제어할 수 있어야 한다.
- 기기와 소비자 간의 GPS 정보 기반 거리가 10m 이하일 때만 잠금 해제가 가능하다.
- 소비자가 기기의 잠금 해제를 진행하면 기기의 상태가 배달 완료로 변경되며 가게로 배달 완료 알림이 전송된다.
- 가게는 카메라를 사용할 수 있어야 한다.

5. 요구사항 목록

서브 컴포넌트	요구사항 ID	요구사항
ID	ID01-01	기기 식별 정보 정의
	ID01-02	가게 식별 정보 정의
	ID01-03	소비자 식별 정보 정의
	ID01-04	주문 식별 정보 정의
Configuration	CF02-01	시스템 모듈 Configuration
	CF02-02	BackEnd SW Configuration
	CF02-03	기기 SW Configuration
	CF02-04	가게 SW Configuration
	CF02-05	소비자 SW Configuration
Communication	CM03-01	시스템 모듈 간 연결
	CM03-02	프로토콜 정의
	CM03-03	서버-DB 통신
	CM03-04	IoT 허브-앱 서버 통신
	CM03-05	서버-기기 통신
	CM03-06	서버-가게 통신
	CM03-07	서버-소비자 통신
Device	DV04-01	GPIO 설정
	DV04-02	배터리 잔량 정보
	DV04-03	GPS 정보
	DV04-04	주문 할당 LED 제어
	DV04-05	잠금장치 제어
	DV04-06	스피커 제어
	DV04-07	시스템 오류 관리
	DV04-08	실행 스크립트
Store	ST05-01	가게 회원 등록
	ST05-02	가게 로그인 / 로그아웃
	ST05-03	주문 접수 목록 조회
	ST05-04	주문 접수 정보 조회
	ST05-05	접수 주문 기기 할당
	ST05-06	접수 주문 기기 할당 해제
	ST05-07	접수 주문 기기 잠금
	ST05-08	접수 주문 기기 잠금 해제
	ST05-09	배달 시작 안내
	ST05-10	배달 완료 알림
Consumer	CS06-01	소비자 회원 등록
	CS06-02	소비자 로그인 / 로그아웃
	CS06-03	가게 목록 조회
	CS06-04	배달 주문 요청
	CS06-05	배달 주문 목록 조회

	CS06-06	배달 주문 정보 조회
	CS06-07	배달 시작 알림
	CS06-08	기기 잠금 해제
BackEnd	BE07-01	IoT 허브 - Layer
	BE07-02	앱 서버 - Layer
	BE07-03	기기 요청 정의
	BE07-04	가게 요청 정의
	BE07-05	소비자 요청 정의
	BE07-06	서버 간 요청

6. 요구사항 명세

6.1 ID

요구사항 ID	요구사항	상세 요구사항
ID01-01	기기 식별 정보 정의	<ul style="list-style-type: none"> ○ 기기 식별 정보는 다음과 같이 정의한다. - dev_id : 기기 고유 식별번호 - state : 기기 상태(배달 준비, 배달 중, 배달 완료) - battery : 기기 배터리 잔량의 25% 기준 여부 - alloc : 기기 주문 할당 여부 - lock : 기기 잠금장치 활성화 여부 - latitude : 기기의 GPS 상 위도 - longitude : 기기의 GPS 상 경도
ID01-02	가게 식별 정보 정의	<ul style="list-style-type: none"> ○ 가게 식별 정보는 다음과 같이 정의한다. - store_id : 가게 계정 ID - store_pw : 가게 계정 비밀번호 - store_name : 가게 이름
ID01-03	소비자 식별 정보 정의	<ul style="list-style-type: none"> ○ 소비자 식별 정보는 다음과 같이 정의한다. - cons_id : 소비자 계정 ID - cons_pw : 소비자 계정 비밀번호
ID01-04	주문 식별 정보 정의	<ul style="list-style-type: none"> ○ 주문 식별 정보는 다음과 같이 정의한다. - order_id : 주문 고유 식별번호 - store_id : 주문이 접수된 가게 ID - cons_id : 주문을 요청한 소비자 ID - dev_id : 주문에 대해 할당된 기기 고유 식별번호 - state : 주문 진행 상태

6.2 Configuration

요구사항 ID	요구사항	상세 요구사항
CF02-01	시스템 모듈 Configuration	<ul style="list-style-type: none"> ○ 시스템 모듈은 다음과 같이 구성한다. - BackEnd : 푸드 박스 백엔드 SW - Device : 푸드 박스 기기 임베디드 SW - Store : 푸드 박스 가게용 모바일 앱 SW - Consumer : 푸드 박스 소비자용 모바일 앱 SW
CF02-02	BackEnd SW Configuration	<ul style="list-style-type: none"> ○ 백엔드 SW는 다음과 같이 구성한다. - IoT 허브 Contoller Layer - IoT 허브 Service Layer - IoT 허브 DAO Layer - IoT DB - 앱 서버 Contoller Layer - 앱 서버 Service Layer - 앱 서버 DAO Layer - 앱 DB ○ 백엔드 SW는 네트워크 클라우드 환경에서 구동한다. ○ 백엔드 SW는 Spring Boot Framework로 개발한다. ○ 백엔드 SW는 각각의 서버 레벨의 포트를 8080으로 지정한다. ○ 백엔드 DBMS는 MariaDB로 지정하며 포트를 3306으로 지정한다.
CF02-03	기기 SW Configuration	<ul style="list-style-type: none"> ○ 기기 SW는 메인 어플리케이션, 실행 Script로 구성한다. ○ 기기 SW는 라즈베리파이 OS에서 구동한다. ○ 메인 어플리케이션은 Python으로 개발한다. ○ 실행 스크립트는 Bash Shell Script로 개발한다. ○ 기기 내의 로컬 파일은 다음과 같다. - 기기 고유 식별번호 파일 - IoT 허브 IP/Port 파일(암호화) ○ 서버 접속을 위한 IP/Port의 복호화가 필요하다. ○ 기기 SW는 이동통신 네트워크에 연결되어 있어야 한다. ○ 기기 SW는 GPS 통신이 가능해야 한다.
CF02-04	가게 SW Configuration	<ul style="list-style-type: none"> ○ 가게 SW는 Android OS, IOS에서 구동한다. ○ 가게 SW는 React Native Framework로 개발한다. ○ 가게 SW는 다음과 같은 기능을 수행할 수 있다. - 회원 등록 - 로그인 / 로그아웃 - 주문 접수 정보 및 목록 조회 - 접수 주문 기기 할당 및 해제 - 접수 주문 기기 잠금 및 해제 - 배달 시작 안내 ○ 기기 내의 로컬 파일은 다음과 같다. - IoT 허브 IP/Port 파일(암호화) ○ 서버 접속을 위한 IP/Port의 복호화가 필요하다.
CF02-05	소비자 SW Configuration	<ul style="list-style-type: none"> ○ 가게 SW는 Android OS, IOS에서 구동한다. ○ 가게 SW는 React Native Framework로 개발한다. ○ 가게 SW는 다음과 같은 기능을 수행할 수 있다. - 회원 등록 - 로그인 / 로그아웃 - 배달 주문 요청

		<ul style="list-style-type: none"> -배달 시작 알림 -기기 잠금 해제 <p>○ 기기 내의 로컬 파일은 다음과 같다.</p> <ul style="list-style-type: none"> -IoT 허브 IP/Port 파일(암호화) <p>○ 서버 접속을 위한 IP/Port의 복호화가 필요하다.</p>
--	--	---

6.3 Communication

요구사항 ID	요구사항	상세 요구사항
CM03-01	시스템 모듈 간 연결	<p>○ 각 시스템 모듈 간의 연결은 다음과 같다.</p> <ul style="list-style-type: none"> - IoT 허브 : IoT DB - 앱 서버 : 앱 DB - IoT 허브 : 앱 서버 - IoT 허브 : 기기 - 앱 서버 : 가게 - 앱 서버 : 소비자
CM03-02	프로토콜 정의	<p>○ 백엔드(IoT 허브, 앱 서버)에서 클라이언트로의 요청에 대한 통신은 WebSocket TCP를 이용한다.</p> <p>○ 클라이언트에서 백엔드로의 요청, 서버 간 요청에 대한 통신은 HTTP를 이용한다.</p>
CM03-03	서버-DB 통신	<p>○ IoT 허브와 앱 서버가 각각 IoT DB와 앱 DB와 통신을 할 때 DAO(Data Access Object)를 이용한다.</p> <p>○ 앱 서버에서 IoT DB의 데이터가 필요한 경우, IoT 허브로 데이터를 요청한다.</p>
CM03-04	IoT 허브-앱 서버 통신	<p>○ Request를 하는 경우 HTTP 통신으로 진행한다.</p> <p>○ 앱 서버에서 IoT 허브 방향으로의 Request만 발생하고 반대의 경우는 발생하지 않는다.</p> <p>○ 앱 서버에서 IoT 허브로 HTTP Request를 전송해야 한다.</p> <p>○ IoT 허브에서 앱 서버의 HTTP Request를 수신하고 Response를 전송해야 한다.</p>
CM03-05	서버-기기 통신	<p>○ 기기 클라이언트가 접속할 때 기기 고유 식별번호를 전달받는다.</p> <p>○ 전달받은 기기 고유 식별번호는 기기 클라이언트 세션 객체에 저장한다.</p> <p>○ 해당 고유 식별번호를 통해 세션을 식별하여 요청한다.</p>
CM03-06	서버-가게 통신	<p>○ 기기 클라이언트가 접속할 때 가게 ID를 전달받는다.</p> <p>○ 전달받은 가게 ID는 기기 클라이언트 세션 객체에 저장한다.</p> <p>○ 해당 가게 ID를 통해 세션을 식별하여 요청한다.</p>
CM03-07	서버-소비자 통신	<p>○ 기기 클라이언트가 접속할 때 소비자 ID를 전달받는다.</p> <p>○ 전달받은 소비자 ID는 기기 클라이언트 세션 객체에 저장한다.</p> <p>○ 해당 소비자 ID를 통해 세션을 식별하여 요청한다.</p>

6.4 Device

요구사항 ID	요구사항	상세 요구사항
DV04-01	GPIO 설정	<ul style="list-style-type: none"> ○ GPIO의 핀 번호 및 I/O 설정은 다음과 같다. <ul style="list-style-type: none"> ● 배터리 잔량 측정 회로 <ul style="list-style-type: none"> - Pin Number : GPIO22 - I/O : INPUT - A/D : Digital ● 주문 할당 여부 LED <ul style="list-style-type: none"> - Pin Number : GPIO19 - I/O : OUTPUT - A/D : Digital ● 도어락 잠금장치 제어 서보모터 <ul style="list-style-type: none"> - Pin Number : GPIO18(PWM0) - I/O : OUTPUT - A/D : Analog ● 상태 변화 안내 스피커 <ul style="list-style-type: none"> - Pin Number : GPIO17 - I/O : OUTPUT - A/D : Digital
DV04-02	배터리 잔량 정보	<ul style="list-style-type: none"> ○ 배터리 잔량이 25% 이상인지 미만인지 DigitalRead한 boolean 데이터 ○ 1분에 1회씩 읽어 IoT 허브로 Post한다.
DV04-03	GPS 정보	<ul style="list-style-type: none"> ○ 기기의 현재 GPS 좌표에 대한 데이터를 1분에 1회씩 읽는다. ○ 데이터를 읽을 때마다 IoT 허브로 Post한다.
DV04-04	주문 할당 LED 제어	<ul style="list-style-type: none"> ○ IoT허브로부터 주문 할당 정보 변동 Request가 들어올 때 LED 제어 <ul style="list-style-type: none"> - 주문 할당 시 점등 - 주문 할당 해제 시 소등
DV04-05	잠금장치 제어	<ul style="list-style-type: none"> ○ IoT허브로부터 잠금장치 제어 Request가 들어올 때 서보모터 제어 <ul style="list-style-type: none"> - 잠금 요청 시, 서보모터 각도 90도 - 잠금 해제 요청 시, 서보모터 각도 0도
DV04-06	스피커 제어	<ul style="list-style-type: none"> ○ IoT 허브로부터 일련의 Request가 들어올 때 각 Request에 대응하는 안내 음성 출력
DV04-07	시스템 오류 관리	<ul style="list-style-type: none"> ○ 일련의 업무(서버 접속, 데이터 송수신, GPIO 제어) 진행 중 오류 발생 시 로그 파일로 작성해 로컬에 저장 <ul style="list-style-type: none"> - 파일 이름 : error_log.txt - 시스템 필요 권한 : 쓰기
DV04-08	실행 스크립트	<ul style="list-style-type: none"> ○ 시스템 구동 파일을 실행하는 스크립트 파일을 /etc/profile.d에 저장한다. <ul style="list-style-type: none"> - 스크립트 파일에서는 이동통신 네트워크, GPS 기능의 가능 여부를 확인 후 시스템 구동 파일 실행

6.5 Store

요구사항 ID	요구사항	상세 요구사항
ST05-01	가게 회원 등록	<ul style="list-style-type: none"> ○ 다음과 같은 정보를 입력하여 회원을 등록한다. <ul style="list-style-type: none"> - ID - 가게명 - 비밀번호 ○ 앱 서버로 입력한 ID를 전송해 ID 중복 체크를 한다. <ul style="list-style-type: none"> - 중복일 경우 중복 안내 팝업을 출력한다. ○ ID 중복이 아닌 경우 앱 서버로 데이터를 입력한 데이터를 전송한다.
ST05-02	가게 로그인 / 로그아웃	<ul style="list-style-type: none"> ○ 다음과 같은 정보를 입력하여 로그인을 한다. <ul style="list-style-type: none"> - ID - 비밀번호 ○ 앱 서버로 입력한 데이터를 전송해 로그인 유효성을 검증받는다. <ul style="list-style-type: none"> - 유효하지 않은 경우 로그인 실패 팝업을 출력한다. - 유효한 경우 주문 접수 목록을 출력한다. ○ 이후 로그아웃하면 다시 로그인을 할 수 있다.
ST05-03	주문 접수 목록 조회	<ul style="list-style-type: none"> ○ 주문 접수 목록의 아이템에서는 다음과 같은 정보를 출력한다. <ul style="list-style-type: none"> - 주문 고유 식별번호 - 주문 할당 여부 ○ 주문 접수 목록의 접수 정보 프리뷰를 통해 주문 접수 정보를 출력하여 조회할 수 있다. ○ 위와 같은 기능에 필요한 정보는 앱 서버를 통해 요청한다. ○ 새로고침을 할 수 있다.
ST05-04	주문 접수 정보 조회	<ul style="list-style-type: none"> ○ 주문 접수 정보를 조회하면 다음과 같은 정보를 출력한다. <ul style="list-style-type: none"> - 주문 고유 식별번호 - 주문을 요청한 소비자 ID - 주문에 할당된 기기 고유 식별번호(아직 할당하지 않은 경우 출력하지 않는다.) ○ 주문 접수 정보에 기기를 할당할 수 있다. ○ 기존에 할당된 기기의 할당을 해제할 수 있다. ○ 배달 시작을 안내할 수 있다. ○ 위와 같은 기능에 필요한 정보는 앱 서버를 통해 요청한다. ○ 새로고침을 할 수 있다.
ST05-05	접수 주문 기기 할당	<ul style="list-style-type: none"> ○ 접수된 주문에 기기 할당을 할 때 카메라를 통해 기기의 QR 코드를 스캔한다. ○ 스캔 이후 서버 측으로 주문에 대한 기기 할당을 요청한다. ○ 기기 할당이 완료되면 주문 접수 정보에 할당된 기기의 고유 식별번호가 표기되며 안내 팝업을 출력한다.
ST05-06	접수 주문 기기 할당 해제	<ul style="list-style-type: none"> ○ 기존에 할당된 기기의 할당을 해제할 때 앱 서버를 통해 기기 할당 해제를 요청한다. ○ 기기 할당 해제가 완료되면 주문 접수 정보에서 할당된 기기의 고유 식별번호가 소멸하며 안내 팝업을 출력한다.
ST05-07	접수 주문 기기 잠금	<ul style="list-style-type: none"> ○ 주문에 할당된 기기의 잠금장치 활성화를 앱 서버에 요청할 수 있다. ○ 잠금 활성화가 완료되면 안내 팝업을 출력한다.
ST05-08	접수 주문 기기 잠금 해제	<ul style="list-style-type: none"> ○ 주문에 할당된 기기의 잠금장치 비활성화를 앱 서버에 요청할 수 있다. ○ 잠금 비활성화가 완료되면 안내 팝업을 출력한다.

ST05-09	배달 시작 안내	○ 배달 시작을 안내할 때 앱 서버에 배달 시작 안내를 요청한다. ○ 배달 시작 안내를 요청하면 할당된 기기의 잠금이 활성화된다.
ST05-10	배달 완료 알림	○ 앱 서버에서 배달 완료 알림 출력 요청을 받으면 해당 알림을 출력한다. ○ 알림의 텍스트는 다음과 같다. - "{소비자 ID}에 배달한 음식의 수량이 완료되었습니다."

6.6 Consumer

요구사항 ID	요구사항	상세 요구사항
CS06-01	소비자 회원 등록	○ 다음과 같은 정보를 입력하여 회원을 등록한다. - ID - 비밀번호 ○ 앱 서버로 입력한 ID를 전송해 ID 중복 체크를 한다. - 중복일 경우 중복 안내 팝업을 출력한다. ○ ID 중복이 아닌 경우 앱 서버로 데이터를 입력한 데이터를 전송한다.
CS06-02	소비자 로그인 / 로그아웃	○ 다음과 같은 정보를 입력하여 로그인을 한다. - ID - 비밀번호 ○ 앱 서버로 입력한 데이터를 전송해 로그인 유효성을 검증받는다. - 유효하지 않은 경우 로그인 실패 팝업을 출력한다. - 유효한 경우 주문이 가능한 가게 목록을 출력한다. ○ 이후 로그아웃하면 다시 로그인을 할 수 있다.
CS06-03	가게 목록 조회	○ 주문이 가능한 가게 목록을 조회한다. ○ 앱 서버로부터 가게 목록을 수신하여 데이터를 획득한다.
CS06-04	배달 주문 요청	○ 주문이 가능한 가게 목록에서 하나의 가게를 선택하여 배달 주문을 요청한다. ○ 주문을 요청할 때 앱 서버로 주문을 요청한 소비자 ID를 전송한다.
CS06-05	배달 주문 목록 조회	○ 배달 주문 목록의 아이템에서는 다음과 같은 정보를 출력한다. - 주문한 가게명 ○ 주문 접수 목록의 접수 정보 프리뷰를 통해 주문 접수 정보를 출력하여 조회할 수 있다. ○ 위와 같은 기능에 필요한 정보는 앱 서버를 통해 요청한다. ○ 새로그침을 할 수 있다.
CS06-06	배달 주문 정보 조회	○ 요청을 완료한 주문에 대해 주문 정보를 조회할 수 있다. ○ 주문에 할당된 기기의 GPS 위치 데이터를 조회할 수 있다. ○ 기기의 잠금 해제를 요청할 수 있다. ○ 새로그침을 할 수 있다.
CS06-07	배달 시작 알림	○ 앱 서버에서 배달 시작 알림 출력 요청을 받으면 해당 알림을 출력한다. ○ 알림의 텍스트는 다음과 같다. - "{가게명}에 주문한 음식의 배달이 시작되었습니다."
CS06-08	기기 잠금 해제	○ 기기 잠금 해제 요청을 할 때 소비자 ID와 현재 GPS 위치 데이터를 앱 서버로 전송한다. - 잠금 해제 불가 시, 기기와의 거리가 멈을 안내하는 팝업 출력 - 잠금 해제 시, 잠금 해제 성공 안내 팝업 출력

6.7 BackEnd

요구사항 ID	요구사항	상세 요구사항
BE07-01	IoT 허브 - Layer	<ul style="list-style-type: none"> ○ IoT 허브의 Layer는 다음과 같은 구조이다. - IoTController : 기기 클라이언트의 요청을 처리하며 Service를 호출한다. - AppController : 앱 서버의 요청을 처리하며 Service를 호출한다. - IoTService : 비즈니스 로직을 수행하며 DAO를 호출한다. - DAO(Data Access Object) : IoT DB에 접속하여 쿼리를 호출한다.
BE07-02	앱 서버 - Layer	<ul style="list-style-type: none"> ○ 앱 서버의 Layer는 다음과 같은 구조이다. - Controller(가게 앱, 소비자 앱 요청) - Service(가게, 소비자, 주문 정보 로직) - DAO(Data Access Object) (가게, 소비자, 주문 정보 Data)
BE07-03	기기 요청 정의	<ul style="list-style-type: none"> ○ 기기에서 IoT 허브로 전송하는 Request 목록은 다음과 같다. ● IoT 허브에 접속한다. ● 배터리 잔량 정보를 전송한다. ● GPS 위치 데이터를 전송한다. ○ IoT 허브에서 기기로 전송하는 Request 목록은 다음과 같다. ● 주문 할당 여부를 전송한다. ● 잠금장치 활성화 여부를 전송한다. ○ 상세 내용은 인터페이스 정의서에서 작성한다.
BE07-04	가게 요청 정의	<ul style="list-style-type: none"> ○ 가게에서 앱 서버로 전송하는 Request 목록은 다음과 같다. ● 회원 등록 시 ID 중복 체크를 요청한다. ● 회원 등록을 요청한다. ● 로그인을 요청한다. ● 로그아웃을 요청한다. ● 기기에 대한 주문 할당을 요청한다. ● 기기에 대한 주문 할당 해제를 요청한다. ● 기기에 대한 잠금장치 활성화를 요청한다. ● 기기에 대한 잠금장치 비활성화를 요청한다. ● 배달 시작 안내를 요청한다. ○ 앱 서버에서 가게로 전송하는 Request 목록 및 메시지, 이후 Request는 다음과 같다. ● 주문 접수 목록을 전송한다. ● 주문 접수 정보를 전송한다. ● 배달 완료 신호를 전송한다. ○ 상세 내용은 인터페이스 정의서에서 작성한다.
BE07-05	소비자 요청 정의	<ul style="list-style-type: none"> ○ 소비자에서 앱 서버로 전송하는 Request 목록 및 메시지, 이후 Request는 다음과 같다. ● 회원 등록 시 ID 중복 체크를 요청한다. ● 회원 등록을 요청한다. ● 로그인을 요청한다. ● 로그아웃을 요청한다. ● 배달 주문을 요청한다. ● 기기 잠금 해제를 요청한다.

		<p>○ 앱 서버에서 소비자로 전송하는 Request 목록 및 메시지, 이후 Request는 다음과 같다.</p> <ul style="list-style-type: none"> ● 주문 가능한 가게 목록을 전송한다. ● 주문 목록을 전송한다. ● 주문 정보를 전송한다. ● 배달 시작 알림을 전송한다. <p>○ 상세 내용은 인터페이스 정의서에서 작성한다.</p>
BE07-06	서버 간 요청	<p>○ 앱 서버에서 IoT 허브로 전송하는 Request 목록 및 메시지, 이후 Request는 다음과 같다.</p> <ul style="list-style-type: none"> ● 기기의 GPS 위치 데이터를 요청한다. ● 기기의 주문 할당을 요청한다. ● 기기의 주문 할당 해제를 요청한다. ● 기기의 잠금장치 활성화를 요청한다. ● 기기의 잠금장치 비활성화를 요청한다. <p>○ 상세 내용은 인터페이스 정의서에서 작성한다.</p>