

# Baskin Robbins Kiosk



JAVA기반 스마트 웹&앱 개발자 양성

Team. CREAM

김푸름 박경호 박준영 백지수 최은지

# 목차

## 01 SCHEDULE p.4-5

- 진행 일정

## 02 CRITICISM p.7

- 기존 시스템에 대한 비판

## 03 PERSONA p.9-13

- 니즈에 따른 해결 방안

## 04 DESIGN p.15-22

- 드로잉  
- 구현

## 05 CODE p.24-38

- 사용 도구  
- 코드

## 06 END p.40-44

- REVIEW



# 01

---

# SCHEDULE



# SCHEDULE

진행 일정 (10.14 - 10.26)

10.14

페르소나,  
기존 기능 비판,  
필요 기능 및  
개선 기능 정리

10.15

화면 설계서,  
기능 명세서  
초안 작성

10.16 - 10.18

문서 체계화  
(화면 설계서,  
기능 명세서,  
Wire Frame)

10.19 - 10.20

세부 디자인 및  
컨셉트 작업,  
기본 코드  
밑 작업



# SCHEDULE

진행 일정 (10.14 - 10.26)

10.21 - 10.22

Front-end  
세부 작업,  
Back-end  
작업 진행

10.23 - 10.24

오류 수정,  
코드 Confirm

10.25

발표 문서 작업

10.26

최종 점검,  
프로젝트 종료



## 02

---

# CRITICISM

# CRITICISM

기준 시스템에 대한 비판

## 상품에 대한 불충분한 안내

아이스크림 등 상품을 선택한 뒤 나타나는 화면에 아이스크림의 맛, 알레르기에 대한 정보가 충분하지 않음

## 기준을 알 수 없는 상품 정렬

신메뉴, 인기 메뉴 등이 최상단에 노출되지 않고 이리저리 산재되어 있어 사용자로 하여금 혼란스러움을 느끼게 함

## 잔여시간의 비가시성

잔여 시간이 화면에 표시되지 않음. 종료 10초 전에 종료 안내창을 띄워주기는 하지만 텍스트의 사이즈가 작고, 양이 많아 짧은 시간 내에 읽기 어려움

## 불필요한 동작 존재

메인 화면에서 메뉴 선택 화면으로 넘어 가기 전, 불필요하게 메뉴의 카테고리를 한번 더 선택하는 동작 필요

## 눈에 띠지 않는 상호작용

버튼을 클릭해도 눈에 잘 보이는 반응이 일어나지 않아 신체적인 특성을 가진 일부 사용자가 이용에 불편함을 겪음

## 화면 순서의 부적절함

결제하기와 그 다음 단계인 카드 결제 버튼이 같은 곳에 위치. 두 단계 사이에 나타나는 포인트 적립/사용 팝업창이 닫히는 동작 때문에 딜레이가 발생하여 결제하기 버튼을 한 번 더 누르다가 본의 아니게 카드 결제 버튼을 클릭하게 됨



# 03

---

# PERSONA

# PERSONA



**이름:** 박수현

**나이:** 42세(만 41세)

**직업:** 저작권 관리원

**취미:** 사이클, 소설 쓰기

**거주:** 서울특별시 용산구

**성격:** 차분하지만, 가끔 너무 웃길 때면 까마귀 같은 소리를 내며 웃어서 주위 사람들을 놀라게 한다. 스트레스를 덜 받고, 자주 웃는 편. 할 말은 다 하는 성격이다.



## 최근 관심사

딸의 초등학교 생활이 최대 관심사이다. 아이와 더 잘 지내기 위해 2세대 아이들이 무엇을 좋아하는지, 무엇을 입는지, 무엇을 먹는지 알려주는 레터를 구독해서 보고 있다.

## 고민

딸, 아영이가 배스킨라빈스에서 나온 신메뉴를 먹어보고 싶다고 했는데, 키오스크로 주문하기가 너무 어렵다. 색을 구분할 수 없는 완전색 맹 장애를 갖고 있는데, 키오스크 주문을 할 때 버튼이 눌렸는지 확인이 안 돼서 이상한 것을 누르다가 결국 주문을 못 하고 나왔다.

## 니즈

- 버튼을 누르면 눌렸다는 표시가 확실하게 드러났으면 한다.
- 아이스크림 색깔로는 맛을 구분할 수 없어서 맛 설명이 있기를 원한다.

# PERSONA



**이름:** 강유리

**나이:** 18세(만 17세)

**직업:** 소하고등학교 재학생

**취미:** 친구들과 인생네컷 찍기, 방송댄스

**거주:** 경기도 광명시 소하동

**성격:** 평소 사교성이 좋아 친구들과 어울려 노는 것을 즐기며, 좋은 것과 싫은 것에 대한 구분이 분명하다. 밝은 겉모습과는 다르게 내심 타인에 비해 뒤처질까 늘 고민이다.



## 최근 관심사

평소 유행에 민감하게 반응하고, 새로운 것을 좋아하는 성격이기에 편의점이나 각종 가게에 출시되는 캐릭터 콜라보 제품들에 크게 관심을 가지고 있다. 특히 빵이나 아이스크림 종류에는 더 눈이 간다.

## 고민

신제품 출시 소식을 듣고 배스킨라빈스에 찾아갔는데, 메인 화면에서만 신제품 정보가 나오고 맛을 고르는 화면에서는 신메뉴가 무엇인지 알아보기 어려웠다. 또한 인기 메뉴가 중구난방으로 정렬되어 있어 스스로 핸드폰을 꺼내 정보를 검색해보고 주문하는 수밖에 없었다.

## 니즈

- 메뉴 선택 화면에서도 신메뉴를 한눈에 알아볼 수 있었으면 한다.
- 가장 인기있는 메뉴가 무엇인지 직접 찾아보지 않고도 알 수 있도록 순서대로 정렬되어 있기를 원한다.

# PERSONA



**이름:** 이재현

**나이:** 34세(만33세)

**직업:** 자동차 프로모션 운영팀 대리

**취미:** 풋살, 헬스, 맥방 유튜브 시청

**거주:** 충청남도 천안시 동남구

**성격:** 활동적이며 새로운 것을 좋아하지만  
주의력이 산만한 편이다.



## 최근 관심사

먹방을 좋아해서 여러 종류의 음식을 먹는 것에 끝 빠진 상태이다. 평소  
자주 이용하는 헬스장의 1층 상가에 디저트 가게가 생긴다는 소식을 듣고  
기분이 좋다.

## 고민

새로운 맛을 체험하는 것을 좋아하지만 각종 알레르기가 있어서 먹지  
못 하는 식품이 많다. 방문할 때마다 직원에게 원재료에 대하여 물어보  
기에는 직원이 바빠보이고 매장 안에서 원재료 표를 찾아보지만 보이  
지 않는다.

## 니즈

- 원재료의 정보(알레르기 성분 표시)를 키오스크에서 확인하기를 원한다.
- 잔여 시간이 표시되지 않고 처음 화면으로 넘어가 불편하다.
- 결제 진행 과정에서 다음으로 넘어가는 버튼이 잘 보이지 않아 불편하다.

# PERSONA



**이름:** 김서아

**나이:** 28세(만 27세)

**직업:** 송파구청 공무원

**취미:** 등산, 여행, 맛집 탐방

**거주:** 서울특별시 송파구

**성격:** MBTI ESFP, 좋고 싫음이 분명하게 갈려 싫어하는  
사람 앞에서 싫은 티가 잘 나타난다. 매우 활발하여  
친구들과 자주 어울린다.



## 최근 관심사

인스타그램을 채워가는 것에 관심이 있다. 친구들과 새로운 맛집에 다니면서 음식 사진을 찍거나 주말에 등산 모임에서 다 같이 사진을 찍는 일 이 잦기 때문에 아직도 업로드하지 않은 사진이 많다.

## 고민

친구들과 술을 마시고 난 뒤 종종 배스킨라빈스를 찾아 아이스크림을 먹는데, 술에 취한 상태라 키오스크로 주문하기 복잡하다. 특히 느린 반응속도 때문에 주문하는 동안 몇 번이나 실수로 다른 버튼을 터치한 적이 많다.

## 나이즈

- 불필요한 선택창 없이 시작 화면에서 바로 여러 종류의 메뉴를 보여 주었으면 한다.
- 실수로 다른 버튼을 누르지 않도록 확실한 구분이 있었으면 한다.

# PERSONA

## NEEDS & RESULT

### NEEDS

맛에 대한 설명,  
원재료와 알레르기에  
대한 정보

신체적 특성에  
구애받지 않는  
접근성 높은 UI

의미 있는 기준으로  
메뉴를 정렬

불필요한 메뉴와  
동작의 최소화

키오스크 사용  
잔여 시간을  
알 수 있는 수단



### SOLUTION

맛 선택 시 하단에  
맛에 대한 설명과  
원재료 및 알레르기  
성분 표시

버튼 클릭 색상  
변화를 강조,  
순서에 따른  
색상 강조

정렬의 기준을  
1순위 신메뉴,  
2순위 판매량으로  
설정

의미 없는 메뉴  
선택 화면을 제거

잔여 시간 종료  
10초 전 팝업  
알림 표시



# 04

---

# DESIGN

## POINT!



### 눈에 잘 들어오는 화면의 구성요소

외형적인 부분에서 심미성을 해치지 않으면서도 사용자가 불편함을 느끼지 않을 수 있도록 적절한 텍스트 크기를 설정하고, 명암의 대비를 통해 요소의 가시성을 증대



### 색상과 폰트의 통일성, 조화

다양한 색상과 폰트를 사용하기보다는, 레퍼런스로 선정한 기업의 이미지를 대표할 수 있는 것으로 선택하여 조화롭게 적용. 용도에 맞는 색상을 미리 정해두어 통일성을 이룸



### 철저한 계획과 사전 준비 및 소통

여러 명의 인원이나누어 작업하는 만큼, 혹여나 오차가 발생하여 위험에 느껴지지 않도록 드로잉 단계에서부터 많은 회의를 거쳐 철저하게 요소들의 위치와 크기를 설정

# DESIGN 드로잉



**1 화면을 터치해주세요.**

**2**

광고

탭메뉴제목 탭메뉴제목 탭메뉴제목

광구가 좋아하는 외계인의 바나나킥 ₩ 30,000  
광구가 좋아하는 외계인의 바나나킥 ₩ 30,000  
광구가 좋아하는 외계인의 바나나킥 ₩ 30,000  
광구가 좋아하는 외계인의 바나나킥 ₩ 30,000

광구가 좋아하는 외계인의 바나나킥 ₩ 30,000  
광구가 좋아하는 외계인의 바나나킥 ₩ 30,000  
광구가 좋아하는 외계인의 바나나킥 ₩ 30,000  
광구가 좋아하는 외계인의 바나나킥 ₩ 30,000

광구가 좋아하는 외계인의 바나나킥 ₩ 30,000  
광구가 좋아하는 외계인의 바나나킥 ₩ 30,000  
광구가 좋아하는 외계인의 바나나킥 ₩ 30,000  
광구가 좋아하는 외계인의 바나나킥 ₩ 30,000

장바구니 10 ₩300,000,000 결제하기 >

X  
글자 수 제한 이 정도 두줄까지 가능하게  
고른 맛 종류1, 고른 맛 종류2, 고른 맛 종류3고 이상…

**3**

옵션 선택 맛 선택

해피 버스데이 케이크 ₩ 26,000  
레트로 감성의 크림 아이싱과 화려한 색상, 레터링 초콜릿 장식으로 모두의 생일을 특별하게 만들어 줄 케이크

포장  
포장 안함  
아이스 포장 (\*최대 60분)  
보냉백 포장 (\*최대 90분)

스푼  
스푼 없음  
스푼 추가 (일회용)  
99

초  
큰 초 (\*최대 10개)  
작은 초 (\*최대 10개)  
10

< 이전 화면 담기 >

해피 버스데이 케이크  
칼로리 정보 | 알레르기를 발생시킬 수 있는 내용내용내용  
레트로 감성의 크림 아이싱과 화려한 색상, 레터링 초콜릿 장식으로 모두의 생일을 특별하게 만들어 줄 케이크

## 1 홈화면

- 홈화면에는 총 5장의 광고 배너가 슬라이드
- 클릭하면 메뉴 선택 화면으로 이동

## 2 상품안내화면

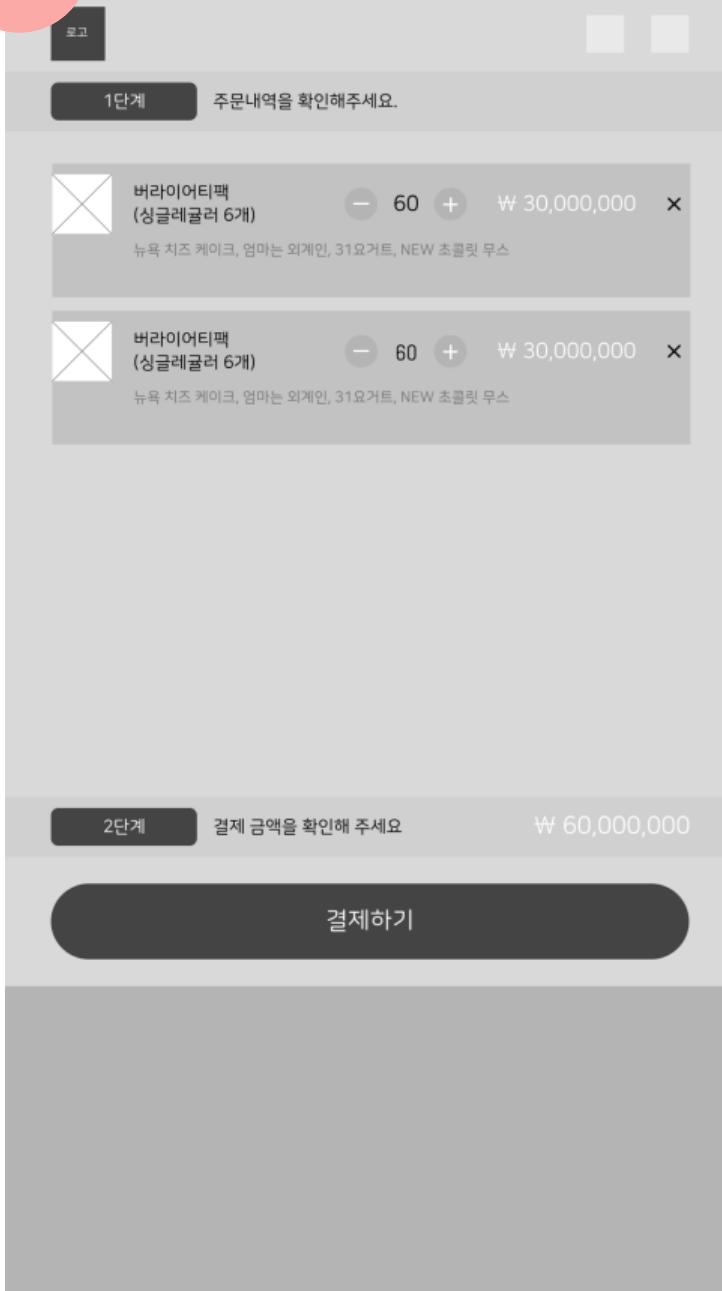
- 상단의 탭을 통해 원하는 항목으로 전환 가능
- 메뉴를 클릭하면 해당하는 옵션화면으로 이동

## 3 옵션선택화면

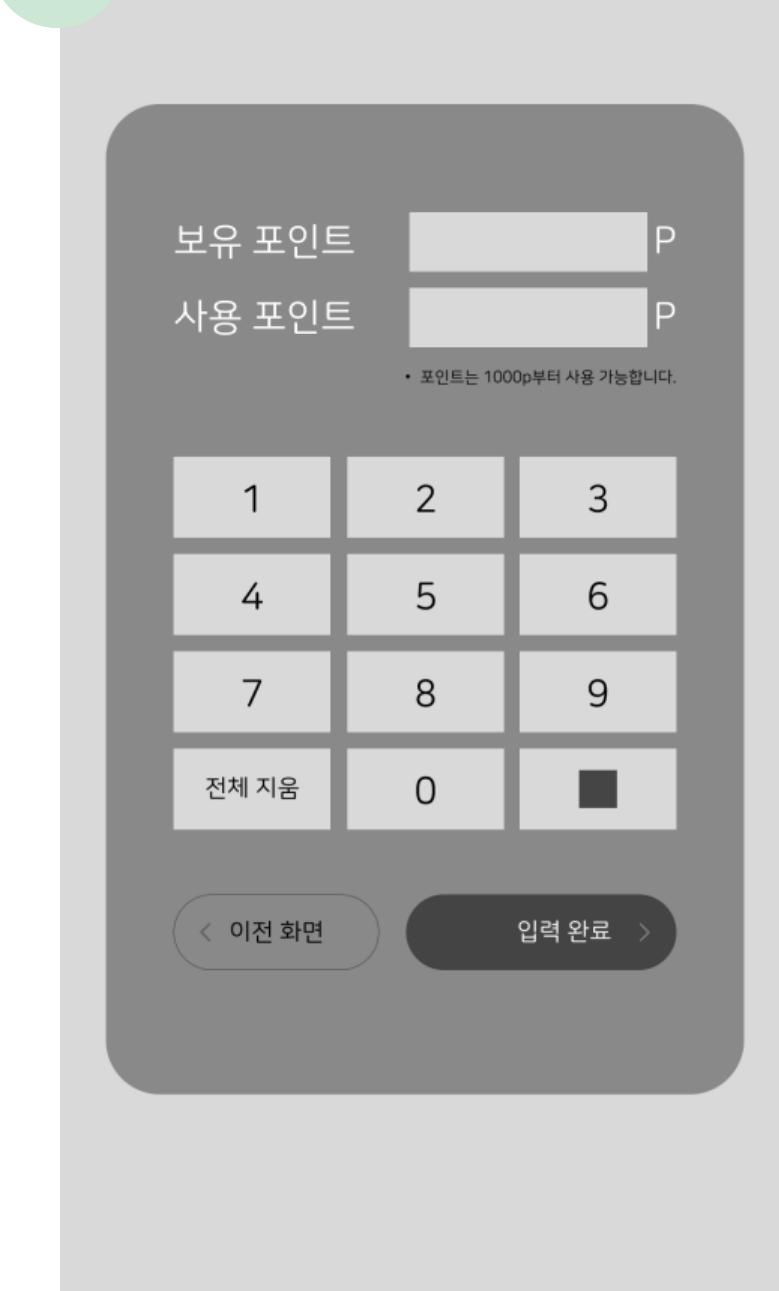
- 선택한 상품에 따른 옵션 설정 가능
- 각 옵션에는 최댓값/최솟값이 설정되어 있음
- 이에 따라 버튼이 활성화되거나 비활성화됨



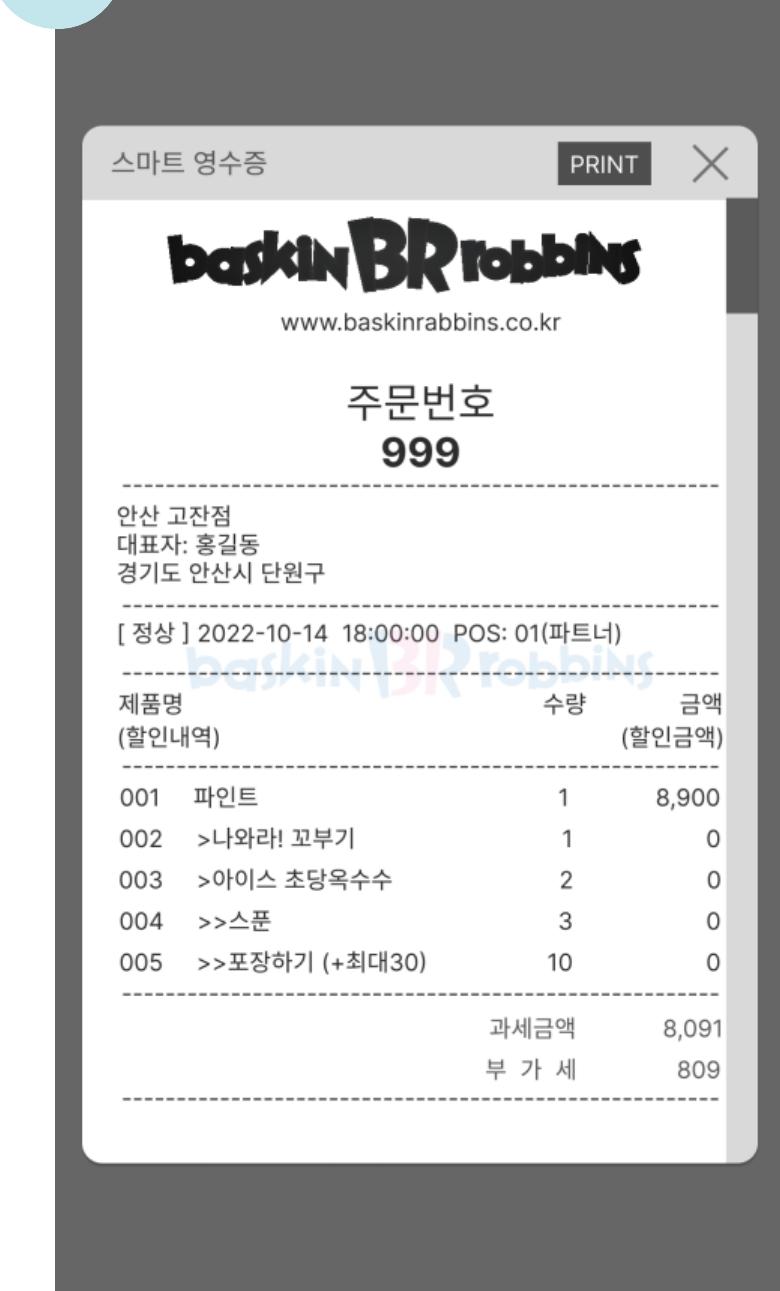
4



5



6



4

## 장바구니 결제 화면

- 사용자가 선택한 모든 상품과 그에 따른 옵션을 확인
- 상품 수량 조절, 삭제 가능

5

## 포인트 사용 화면

- 원하는 만큼 포인트 사용 가능
- 1000P 미만을 입력하거나 보유 포인트 및 총 결제액보다 큰 숫자를 입력할 경우 입력 완료 버튼 비활성화

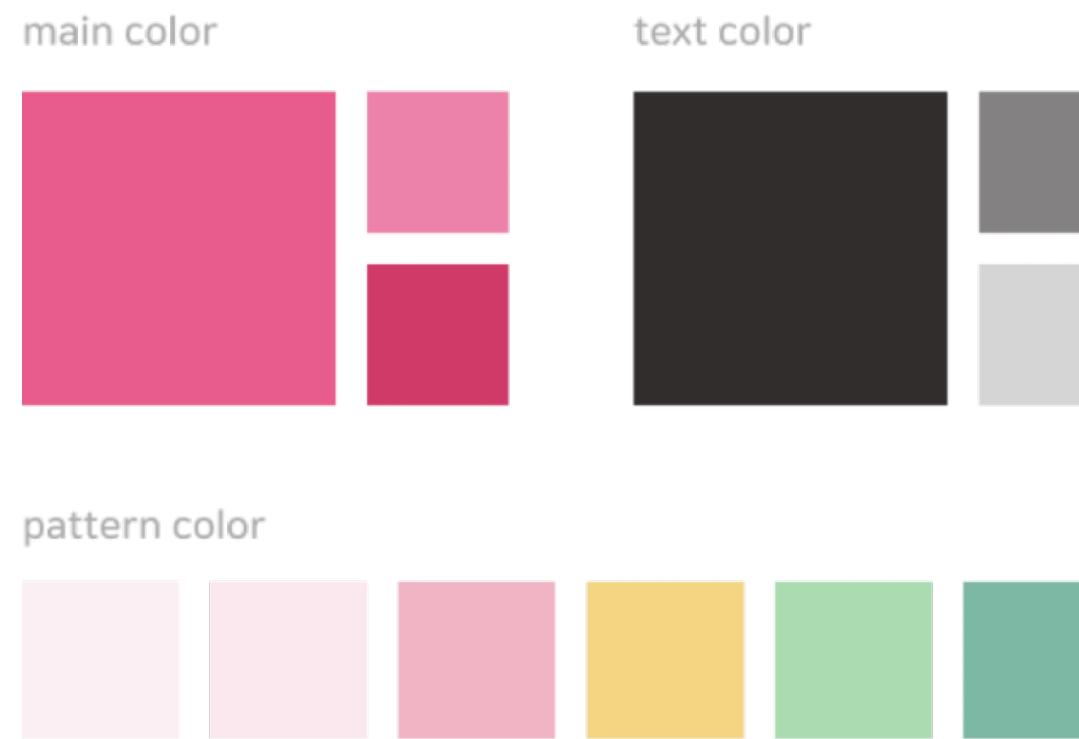
6

## 영수증 확인 화면

- 구매한 상품의 목록, 수량, 가격, 결제 총액 확인
- 상단의 프린트 버튼 클릭 시 영수증 프린트 가능



# VIVID & SWEET



## 나눔스퀘어 라운드

NanumSquareRound

**TITLE**

**BUTTON**

**TEXT**

56px / bold / 1.3  
팝업화면에 들어가는 타이틀 텍스트

34px / bold / 1.3  
버튼에 들어가는 텍스트

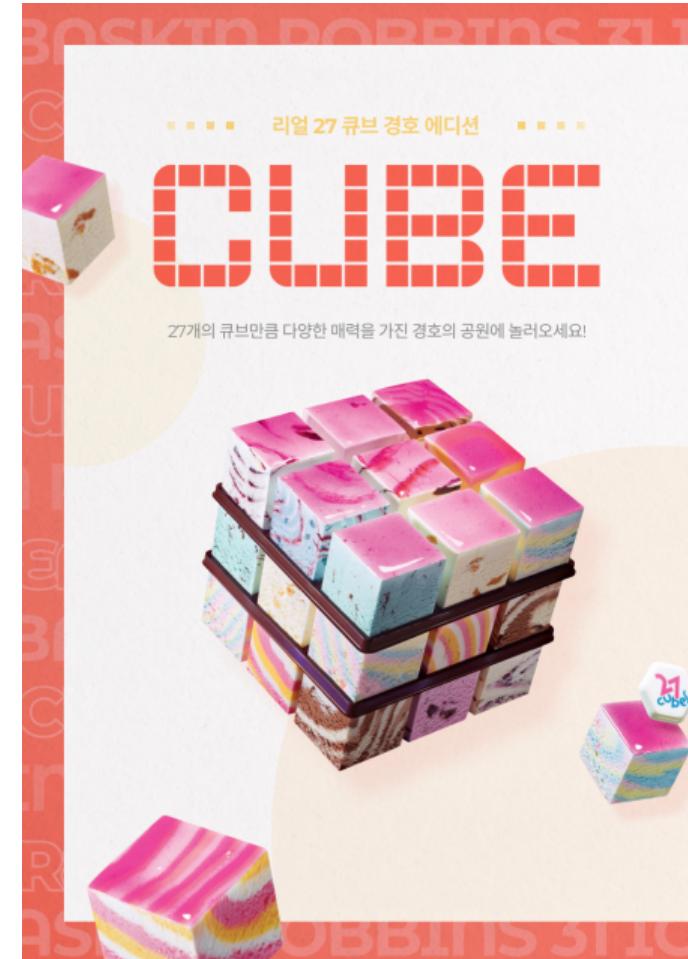
24px / normal, bold / 1.3  
제품 이름, 제품 정보 제목 텍스트



배스킨라빈스가 가지고 있는 이미지를 살린 분홍색 컬러를 메인으로 활용하면서도  
지루함이 느껴지지 않게 다양한 색상을 선정.

요소의 기능과 중요도에 따라 적절하게 배정하고 버튼에 입체감을 주어 귀여우면서  
도 생기 있는 특유의 분위기를 살리도록 디자인

## INDIVIDUALITY



made by. 김푸름

키오스크 페이지에 접속하면 가장 먼저 마주하게 되는 홈 화면에는 아이스크림과 팀원의 포트폴리오 콘셉트를 조화시킨 포스터를 직접 제작해 각자의 개성을 표현.

새로고침 할 때마다 슬라이드의 시작 지점이 달라져 팀원 모두의 다양한 매력을 선보일 수 있도록 페이지를 구성

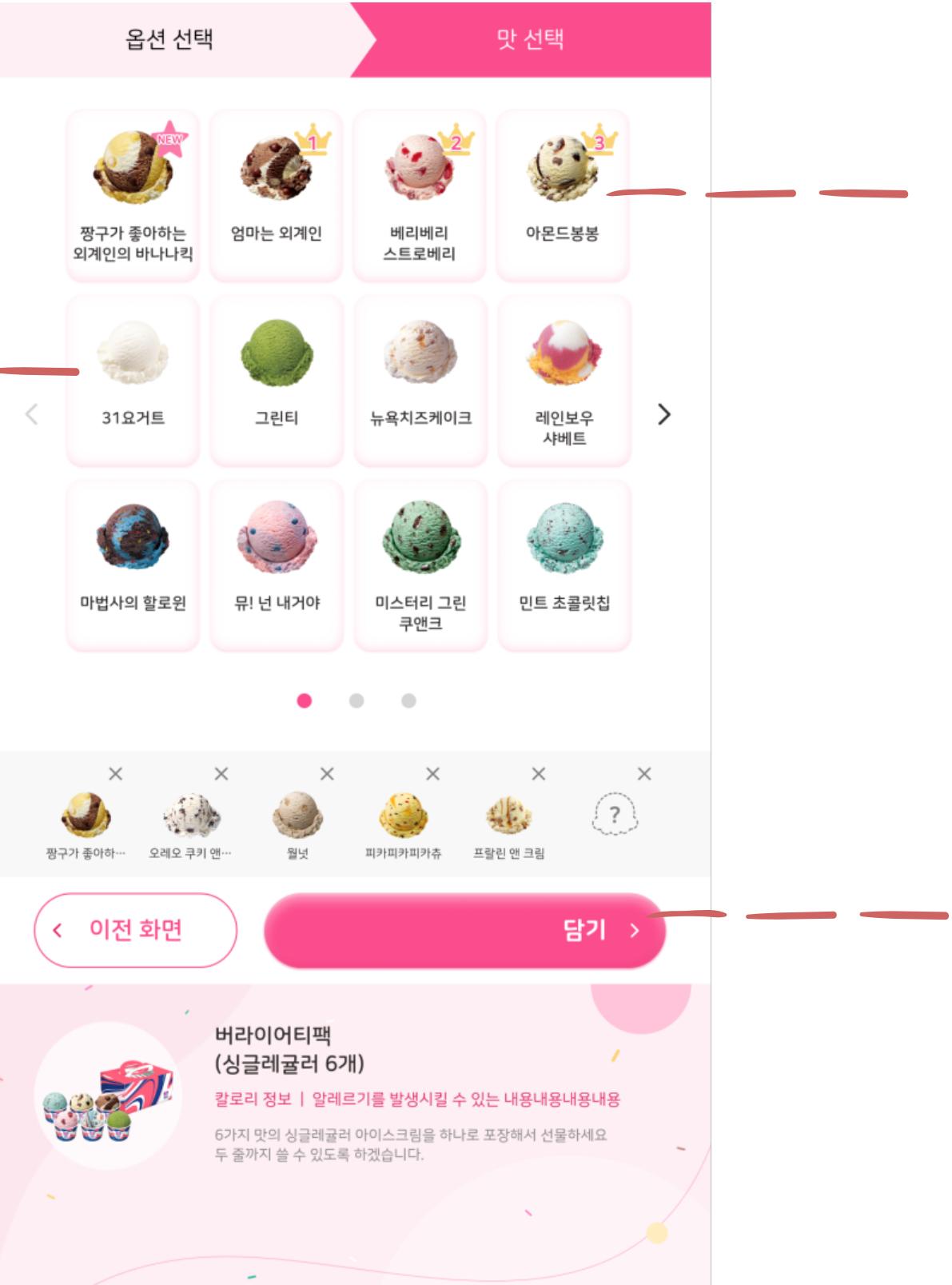


## 눈을 편안하게

시각적인 피로함을 주지 않으면서  
도 다른 제품의 영역과 확실히 구  
분될 수 있도록 디자인된 이미지  
카드

## 제품 정보 표시

제품의 칼로리, 알레르기 등에 대  
한 정보를 하단 영역에 표시하여  
사용자가 원하는 정보를 편리하게  
얻을 수 있음



## 신상품, 인기 상품을 한눈에!

신상품과 인기 상품의 우측 상단에  
아이콘을 장식하여 해당 정보를 원  
하는 사용자의 니즈 충족

## 입체감이 느껴지는 버튼

중요도가 높은 버튼에는 명도와 채  
도가 높은 색을 사용함과 동시에  
입체감을 주어, 버튼의 확실한 구  
분을 원하는 사용자의 니즈 충족



## 올바른 조작을 유도

결제 시 화면에 카드를 투입하는 애니메이션을 표시하여 사용자로 하여금 올바른 조작을 할 수 있도록 유도



## 발랄한 느낌의 디자인 요소

서로 이어지는 일련의 과정에서 혼란을 느끼지 않도록 회원 번호 입력 - 결제 완료까지의 화면을 통일성 있게 디자인

## 일관성 있는 디자인

주제로 설정한 기업이 가지고 있는 상큼한 이미지를 살릴 수 있도록, 메인 컬러와 부합하면서도 발랄한 분위기의 이미지 사용

# DESIGN 구현



**2022년**

일 매출 월 매출 매출 순위

아이스크림 커피 케이크

순위	제품명	수량	가격
1위	아이스 아메리카노	78	₩ 218,400
2위	아이스 민트 초코 모카	24	₩ 91,200
3위	아이스 연유 라떼	9	₩ 34,200
4위	민트 초코 모카	7	₩ 26,600
5위	아이스 카페 라떼	7	₩ 23,100
6위	아이스 바닐라빈 라떼	6	₩ 22,800
7위	아메리카노	7	₩ 19,600
8위	카페 라떼	5	₩ 16,500
9위	아이스 아메리카노(L)	3	₩ 10,500
10위	카페31 아메리카노	2	₩ 9,000
11위	바닐라빈 라떼	2	₩ 7,600
12위	카라멜 마끼아또	2	₩ 7,600
13위	콜드브루 오트라떼	1	₩ 4,800

**판매 순위 화면**



05

CODE



## 코드 작성



Visual Studio Code

## 이미지 편집



Photoshop

## 드로잉 및 협업



Figma

## SKILLS



CSS



JavaScript



Node.js



JSON



Sass



PUG



jQuery



MySQL

# CODE 코드



```
let randomArray = [];
let tempNum = 0;

while( randomArray.length < 5 ){
    let overlap = false;
    tempNum = Math.floor(Math.random() * 5);

    if( randomArray.length == 0){
        randomArray.push(tempNum);
    } else{
        $.each(randomArray, (index, item) => {
            if(item == tempNum){
                overlap = true;
            }
        });
    }

    if(!overlap){
        randomArray.push(tempNum);
    }
}
```

홈 화면 배너를 무작위로 노출하기 위한  
랜덤 배열 생성



# CODE 코드



```
let i = 0;
while(i < randomArray.length){
    $(".wrapHomeSlider").append($(".<li>").css("background-image", `url("./build/img/mainbanner${(randomArray[i] + 1)}.jpg)`));
    i++;
}

$(".wrapHomeSlider").append($(".<li>").css("background-image", `url("./build/img/mainbanner${(randomArray[0] + 1)}.jpg)`));

count = 0;
setInterval(() =>{
    count++;

    $(".wrapHomeSlider").css("transform", `translateX(${-1080 * count}px)`).css("transition", "all 0.5s ease-out");
    if(count == 5){
        setTimeout(()=>{
            $(".wrapHomeSlider").css("transition", "none").css("transform", "translateX(0px)");
            count = 0;
        }, 1600)
    }
}, 4000);
```

랜덤 배열의 값대로 이미지를 화면에 노출  
한 후 4초 간격으로 슬라이드



# CODE 코드

```

switch(cartProduct.cartList[i].productJson["category"]){
    case "icecream":
        $('#areaProductCart article:nth-child(${target+1}) div:nth-child(2)').css("background-image", `url(${this.cartList[i].productJson.url})`);
        $('#areaProductCart article:nth-child(${target+1}) .productTitle').html(` ${this.cartList[i].productJson.name}<br>${this.cartList[i].productJson.sub}`);
        $('#areaProductCart article:nth-child(${target+1}) .productDes').text(this.cartList[i].productJson.description);
        break;
    case "coffee":
        $('#areaProductCart article:nth-child(${target+1}) div:nth-child(2)').css("background-image", `url(${this.cartList[i].productJson.url})`);
        $('#areaProductCart article:nth-child(${target+1}) .productTitle').html(` ${this.cartList[i].productJson.name}`);
        $('#areaProductCart article:nth-child(${target+1}) .productDes').text(this.cartList[i].productJson.description);
        break;
    case "cake":
        $('#areaProductCart article:nth-child(${target+1}) div:nth-child(2)').css("background-image", `url(${this.cartList[i].productJson.url})`);
        $('#areaProductCart article:nth-child(${target+1}) .productTitle').html(` ${this.cartList[i].productJson.name}`);
        $('#areaProductCart article:nth-child(${target+1}) .productDes').text(this.cartList[i].productJson.description);
        break;
}
}

class CartProduct{
    constructor(){
        this.cartList = [];
        this.cartIndex = 0;
        this.cartPage = 0;
        this.cartPriceSum = 0;
        this.cartListCount = 0;
    }
    pushProduct(jsonValue, value){
        value.productJson = jsonValue;
        this.cartList.push(value);
    }
}

```

장바구니에 담긴 상품 목록 등을 주고 있는  
상품 선택 화면의 CartProduct 클래스

JSON 데이터를 파싱하여 상품을 카테고리 별로 분류, 장바구니 목록을 변경



# CODE 코드



```

case "containerOptionBtnIce":{
    outputOptionReturn = sendOption("containerTopInfoIce", "containerOptionIce", "wrapOptionFooterIce");
    $("#pageFlavor").css("display", "block");
    flavorCreate();
    break;
}
case "containerOptionBtnCup":{
    outputOptionReturn = sendOption("containerTopInfoCup", "containerOptionCup", "wrapOptionFooterCup");
    $("#pageFlavor").css("display", "block");
    flavorCreate();
    break;
}
function flavorCreate(){
    class FlavorPage{
        constructor(){
            /**@param {json} json파일 파싱 데이터 - 아이스크림*/
            this.dataFlavor = JSON.parse(JSON.stringify(flavor));
            /**@param {int} 페이지네이션 인덱스 */
            this.pagination = 0;
            /**@param {int} 페이지네이션 수 */
            this.paginationCount = Math.floor(this.dataFlavor.length/12);
            /**@param {object} 선택한 맛 */
            this.flavorSelect;
            /**@param {object[]} 선택한 맛이 담기는 배열*/
            this.flavorSelectArr = [];
            /**
            this.motionBoxList = [];
            // this.flavorIndex = 0;
        }
    }
}

```

맛 선택 화면으로 넘어가야 하는 경우  
flavorCreate() 함수를 실행

로컬 클래스 형태로 만들어둔 Flavor Page가 함수 호출과 동시에 생성



# CODE 코드



```

productClick(){
    // 페이지를 구성하는 함수. 클래스 optionButton, optionExplain 사용
    function optionPageMake(productSelect){
        const{category} = productSelect;
        switch(category){
            case "cake":{
                const {num, name, description, materials, url, price} = productSelect
                const optionButtonCake = new OptionButton($("#containerOptionCake"), "Cake");
                optionButtonCake.makePackageButton();
                optionButtonCake.makeSpoonButton(1);
                optionButtonCake.makeCandleButton();
                let materialsText = "";
                if(materials == null) {
                    materialsText = "준비 중"
                } else {
                    for(let i=0; i<materials.length; i++){
                        if(materials[i] == null) {
                            materialsText += (i+1 != materials.length) ? ('준비 중, ') : ('준비 중');
                        } else {
                            materialsText += (i+1 != materials.length) ? (materials[i] + ', ') : (materials[i]);
                        }
                    }
                }
                const optionInfoCake= new optionExplain(url, `#${name}`, "₩ " + (price.toLocaleString()), description, materialsText);
                optionInfoCake.optionExplainTop("containerTopInfoCake");
                optionInfoCake.optionExplainBottom("wrapOptionFooterCake");
                $("#pageOptionCake").css("display", "block");
                break;
            }
        }
    }
}

```

네 가지의 옵션 페이지를 만들기 위해 선언한 productClick 메소드

상하단의 상품 정보에 들어갈 내용을 만들어내고 화면에 표시하기 위한 코드

옵션 선택
담기

리얼 27 큐브 ₩ 28,000



9가지 맛 아이스크림 바이트를 초콜릿 판 위에 쌓아올려 전체 다 취식 가능한 실속있는 케이크

포장
스푼
초

포장 안함

-
10
+

아이스 포장 (\*최대 60분)

-
10
+

보냉백 포장 (\*최대 90분)

-
10
+

스푼 없음

-
1
+

스푼 추가 (일회용)

-
1
+

큰 초 (\*최대 15개)

-
0
+

작은 초 (\*최대 15개)

-
0
+

이전 화면
담기

리얼 27 큐브



체리쥬빌레, 뉴욕 치즈케이크, 레인보우 사베트, 슈팅스타, 민트 …  
9가지 맛 아이스크림 바이트를 초콜릿 판 위에 쌓아올려 전체 다 취식 가능한 실속있는 케이크



distructuring을 통하여 객체 형태의 정보를 손쉽게 변수에 저장하여 사용

```

case "icecream":{
  const {num, name, sub, description, kcal, url, price, unit, form} = productSelect;
  if(form == "conecup"){
    const optionButtonCup = new OptionButton($("#containerOptionCup"), "Cup");
    optionButtonCup.makeCupButton();
    optionButtonCup.makeSpoonButton(1);
    const optionInfoCup = new optionExplain(url, `${name}<br>${sub}`, "₩ "+(price.toLocaleString()), description, `${kcal}kcal`);
    optionInfoCup.optionExplainTop("containerTopInfoCup");
    optionInfoCup.optionExplainBottom("wrapOptionFooterCup");
    $("#pageOptionCup").css("display", "block");
  } else {
    const optionButtonIce = new OptionButton($("#containerOptionIce"), "Ice");
    optionButtonIce.makePackageButton();
    optionButtonIce.makeSpoonButton(unit);
    const optionInfoIce = new optionExplain(url, `${name}<br>${sub}`, "₩ "+(price.toLocaleString()), description, `${kcal}kcal`);
    optionInfoIce.optionExplainTop("containerTopInfoIce");
    optionInfoIce.optionExplainBottom("wrapOptionFooterIce");
    $("#pageOptionIce").css("display", "block");
  }
  break;
}

```

switch 문으로 상품의 카테고리를 분류하여 상황에 따라 필요한 코드를 실행

옵션 선택
담기

리얼 27 큐브
₩ 28,000

포장
스푼
초

이전 화면
담기

9가지 맛 아이스크림 바이트를 초콜릿 판 위에 쌓아올려 전체 다 취식 가능한 실속있는 케이크

<span style="color: #ff6699;">+</span> 10	<span style="color: #ff6699;">-</span>
-------------------------------------------	----------------------------------------

<span style="color: #ff6699;">+</span> 10	<span style="color: #ff6699;">-</span>
-------------------------------------------	----------------------------------------

<span style="color: #ff6699;">+</span> 1	<span style="color: #ff6699;">-</span>
------------------------------------------	----------------------------------------

<span style="color: #ff6699;">+</span> 0	<span style="color: #ff6699;">-</span>
------------------------------------------	----------------------------------------

<span style="color: #ff6699;">+</span> 0	<span style="color: #ff6699;">-</span>
------------------------------------------	----------------------------------------

30

# CODE 코드



```

const makeCartProductList = ()=>{
  $("#cartListArea").empty();
  cartTotalPrice = 0;
  cartList.forEach(productList, productIndex) => {
    const flavorArr = [];
    if(productList.productJson.category=="icecream"){
      const {url, name, sub, optionJson, price} = productList.productJson;
      productList.forEach(flavorList =>{
        flavorArr.push(flavorList.name);
      });
      cartProductArr.push(new ShoppingCartProduct(url, name, sub, optionJson, flavorArr, price, productIndex));
    }
    else if(productList.productJson.category=="coffee"){
      const {url, name, optionJson, price} = productList.productJson;
      cartProductArr.push(new ShoppingCartProduct(url, name, "", optionJson, [], price, productIndex));
    }
    else if(productList.productJson.category=="cake"){
      const {url, name, materials, optionJson, price} = productList.productJson;
      cartProductArr.push(new ShoppingCartProduct(url, name, "", optionJson, materials, price, productIndex));
    }
    cartProductArr[productIndex].makeBox();
    cartTotalPrice += cartProductArr[productIndex].calcTotalPrice();
  };
  checkCartBox();
  $("#cartTotalPrice").text(cartTotalPrice.toLocaleString());
}

```

이전 화면에서 넘어온 데이터를 파싱하여 forEach를 통해 객체 생성, 배열에 저장

상품	옵션	수량	가격
싱글레귤러 (콘 / 컵)	엄마는 외계인, 콘 / 컵: 콘   스푼: 스푼 없음	1	₩ 3,500
아메리카노	포장: 포장 안함   샷: 샷 추가 없음   시럽: 시럽 추가 없음	1	₩ 2,800
아이스 카페 라떼	포장: 포장 안함   샷: 샷 추가 없음   시럽: 시럽 추가 없음	1	₩ 3,300
특별한 너를 위한 케이크	뉴욕 치즈케이크, 포장: 포장 안함   스푼: 스푼 없음   초: 큰 초0개   작은 초0개	1	₩ 32,000

결제 금액을 확인해 주세요. ₩ 46,400

결제하기

생성된 객체의 메소드를 실행하여 화면에 각 상품 리스트 출력 및 총 결제 금액 계산

# CODE 코드



```
$("#cartListArea .xi-minus, #cartListArea .xi-plus").on("click", function(){
    const productId = $(this).closest("article").index();
    switch($(this).attr("class")){
        case "xi-plus":
            cartProductArr[productId].plusProductAmount($(this));
            break;
        case "xi-minus":
            cartProductArr[productId].minusProductAmount($(this));
            break;
    }
    cartTotalPrice = 0;
    cartProductArr.forEach(productObj=>{
        cartTotalPrice += productObj.calcTotalPrice();
    });
    $("#cartTotalPrice").text(cartTotalPrice.toLocaleString());
});
```

```
$("#btnCartPay").on("click", ()=>{
    resultProductInfoArr = [];
    cartProductArr.forEach(productInfo=>{
        resultProductInfoArr.push(productInfo.makeProductInfo());
    });
    gsap.set(`#pageCart`[0], { display: 'none' });
    gsap.set(`#pageOptionPoint`[0], { display: 'block' });
});
```

'+', '-' 버튼을 눌렀을 때 해당하는 상품의 수량을 증감시킬 수 있는 이벤트를 생성

사용자가 결제하기 버튼을 클릭했을 때, 장바구니에 담긴 정보를 배열에 저장

서버: localhost:3306 » 데이터베이스: c16st10 » 테이블: KioskAdmin

#	이름	종류	데이터정렬방식	보기	Null	기본값	설명	추가	실행
1	<b>id</b> 	int(11)		아니오	없음	AUTO_INCREMENT	 변경	 삭제	 더보기
2	<b>category</b>	varchar(50)	utf8mb4_general_ci	예	NULL		 변경	 삭제	 더보기
3	<b>menu</b>	varchar(100)	utf8mb4_general_ci	예	NULL		 변경	 삭제	 더보기
4	<b>count</b>	int(11)		예	NULL		 변경	 삭제	 더보기
5	<b>price</b>	bigint(20)		예	NULL		 변경	 삭제	 더보기
6	<b>date</b>	datetime		예	NULL		 변경	 삭제	 더보기

KioskMember : 회원 목록 테이블

서버: localhost:3306 » 데이터베이스: c16st10 » 테이블: KioskMember

#	이름	종류	데이터정렬방식	보기	Null	기본값	설명	추가	실행
1	<b>id</b> 	int(11)		아니오	없음	AUTO_INCREMENT	 변경	 삭제	 더보기
2	<b>number</b> 	varchar(11)	utf8mb4_general_ci	아니오	없음		 변경	 삭제	 더보기
3	<b>point</b>	int(11)		예	NULL		 변경	 삭제	 더보기
4	<b>count</b>	int(11)		예	NULL		 변경	 삭제	 더보기
5	<b>date</b>	datetime		예	NULL		 변경	 삭제	 더보기

KioskAdmin : 매출 목록 테이블

# CODE 코드



```
app.get('/kiosk', (req, res)=>{
  con.query('SELECT * FROM KioskAdmin', (err, result, field)=>{
    fs.writeFileSync('./src/adminList.json', JSON.stringify(result));
  })

  let selectMember = 'SELECT * FROM KioskMember';
  let countMax = 0;
  let targetId = 1;
  let dayMax = '';
  let callCount = `SELECT count 'max', id FROM KioskMember WHERE count = (SELECT MAX(count) FROM KioskMember)`;
  con.query(selectMember, (err, result, field)=>{
    fs.writeFileSync('./src/memberList.json', JSON.stringify(result));
    con.query(callCount, (err1, result1, field1)=>{
      countMax = result1[0].max;
      targetId = result1[0].id;
      con.query(`SELECT date FROM KioskMember WHERE id = ${targetId}`, (err2, result2, field2)=>{
        dayMax = result2[0].date;
        res.render('kioskPage', {maxCount: countMax, dayTarget: dayMax});
      })
    })
  })
}

app.post('/kiosk', (req, res, next)=>{
  let userId = req.body.id;
  let userNumber = req.body.number;
  let userPoint = req.body.point;
  let userCount = req.body.count;
  let userDate = req.body.date;
  let keyCount = req.body.keyCount;
  let userKey = req.body.key;
```

관리자 화면의 매출 정보를 출력을 위해  
KioskAdmin Table 값 JSON으로 저장

KioskMember Table에서 회원 목록  
및 주문 번호 출력 위해 당일 마지막 회원  
id의 정보 Pug 변수 전달

Body-Parser를 통해 받은 값의 변수 선언

# CODE 코드



```
let resetCount = `UPDATE KioskMember SET count = 0 WHERE id != ${Number(userId)}`;  
if(keyCount == "2"){  
    con.query(resetCount, (err, result, field)=>{  
        })  
}
```

keyCount: 마지막 구매한 회원 날짜와 현재 계산한 날짜가 상이한 경우 “2”를 전달하여 현재 계산한 회원의 주문 번호를 1로 나머지 회원은 0으로 리셋

```
let adminData = req.body.adminDataList;  
let adminList = [];  
adminData = adminData.split("||");  
for(let i=0; i<adminData.length; i++){  
    adminList.push(adminData[i].split(','))  
}  
let adminInsert = `INSERT INTO KioskAdmin (category, menu, count, price, date) VALUES ?`;  
con.query(adminInsert, [adminList], (err, result, field)=>{  
    console.log('----- admin insert -----')  
})
```

영수증 정보를 받아와 2차원 배열로 변경 후 토큰으로 값 전달하여 KioskAdmin 테이블에 레코드 삽입

```
let updateMember = `UPDATE KioskMember SET point = ${userPoint}, count = ${userCount}, date = "${userDate}" WHERE id = ${Number(userId)}`;  
let insertMember = `INSERT INTO KioskMember (number, point, count, date) VALUES ("${userNumber}", ${userPoint}, ${userCount}, "${userDate}")`;  
let sqlMember = (userKey == 0) ? (insertMember) : (updateMember);  
con.query(sqlMember, (err, result, field)=>{  
    res.redirect('/kiosk');  
});  
  
app.listen(30100, function(){  
    console.log("KIOSK DB CONNECTION");  
});
```

신규 회원 & 기존 회원을 구분하여 신규 회원이면 KioskMember Table에 레코드 삽입, 기존 회원이면 해당 회원의 정보 수정 후 완료

회원 정보 클래스에 담긴 값을 Post 전송 위해 각각의 Input TAG에 값 전달

```

function btnNextINFunc(){
    memberInfo.setNumber = inputMemIN.innerHTML;
    let xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function(){
        if(this.readyState == 4 && this.status == 200) {
            let memberList = JSON.parse(this.responseText);
            memberList.forEach((v,i,a)=>{
                if(v.number == memberInfo.getNumber){
                    memberInfo.setPoint = v.point;
                    memberInfo.setId = v.id;
                    checkNumberJSON = 1;
                }
            })
            if(checkNumberJSON != 1){
                memberInfo.setPoint = 1000;
            }
        }
    };
    xmlhttp.open("GET", "./src/memberList.json", false);
    xmlhttp.send();
}

```

회원 번호 입력 시 실행되는 함수. 저장된 회원 목록 JSON Parsing해서 회원 Class에 값 저장 및 신규 & 기존 회원 구분

```

let pageFinishFunc = () => {
    let prevToday = new Date(Date.parse(memberInfo.getTime));
    let prevDay = ('0' + prevToday.getDate()).slice(-2);
    memberInfo.setTime = prevDay;
    timeCount();
}

```

주문 번호 초기화 위해 결제 시간 저장 후 timeCount() 함수를 통해 회원 테이블에서 가져온 마지막 구매 일자와 비교

```

hiddenPoint.value = memberInfo.getPoint - memberInfo.getUsePoint + memberInfo.getSavePoint;
hiddenCount.value = Number(checkCountJSON) + 1;
... 생략 ...
hiddenKey.value = (checkNumberJSON == 0) ? (0) : (1);

```

# CODE 코드



```
let adminData = "";
for(let i=0; i<resultProductInfoArr.length; i++){
    if(i != resultProductInfoArr.length-1){
        adminData += resultProductInfoArr[i].productCategory + ',' + resultProductInfoArr[i].productName + ',' +
        resultProductInfoArr[i].productAmount + ',' + resultProductInfoArr[i].productPrice + ',' + checkDateJSON + '|';
    } else {
        adminData += resultProductInfoArr[i].productCategory + ',' + resultProductInfoArr[i].productName + ',' +
        resultProductInfoArr[i].productAmount + ',' + resultProductInfoArr[i].productPrice + ',' + checkDateJSON;
    }
}
$('#adminDataList').val(`"${adminData}`);

$(btnPrintYes).on('click', function(event){
    orderReceipt.innerHTML = hiddenCount.value;
    timeReceipt.innerHTML = `[ 정상 ] ${checkDateJSON}`;
    $(listReceipt).empty();
    for(let i=0; i<resultProductInfoArr.length; i++){
        let receiptProductNumber = (i < 9) ? ('00' + (i+1)) : ((i < 99) ? ('0' + (i+1)) : (i+1));
        $(listReceipt).append(`<article><article><p>${receiptProductNumber}</p>
<p>${resultProductInfoArr[i].productName}</p></article><article>
<p>${resultProductInfoArr[i].productAmount}</p><p>${resultProductInfoArr[i].productPrice.toLocaleString('ko-KR')}</p></article></article>`);
    }
    realPriceReceipt.innerHTML = (memberInfo.getPayment * 0.9).toLocaleString('ko-KR');
    //... 생략 ...
    lastPointReceipt.innerHTML = `${memberInfo.getSavePoint.toLocaleString('ko-KR')}/${
        ((memberInfo.getPoint)-(memberInfo.getUsePoint)).toLocaleString('ko-KR')}`;
    nextGsap.to(pageFinish, {y: '192rem'})
        .to(pageReceipt, {y: 0, delay: -0.3});
    pageReceiptFunc();
}

$(btnPrintNo).on('click', function(event){
    nextGsap.to(pageCard, {y: '192rem'});
    $(hiddenBtn).click();
})
```

영수증에 출력될 매출 정보를 Express Post에서 배열로 얻기 위해 문자열로 구성하여 값 전달

영수증 출력 버튼 클릭 시 실제 출력될 각 TAG에 값 전달 및 화면 전환

영수증 미출력 클릭 시 POST  
강제 실행하여 화면 전환

```
class AdminTable{
    constructor(productAllJson){'...'}
    timeCheck(){...}
    changeGMT(value){...}
    changeTimeAll(){...}
    pushListAll(dayVal = this.selectDay, monthVal = this.selectMonth){...}
    printListAll(){...}
    appendList(boxVal, listVal, typeVal){...}
    resetPage(){...}
    adminEvent(){...}
    changeListView(){...}
    changePageAdmin(){...}
}
```

매출 정보를 담을 클래스 및 금일, 이번 달, 이번년도의 매출 정보를 카테고리에 맞게 분리 및 정렬 후 각 페이지에 맞게 출력 진행

```
let adminTable;
let xmlhttpAdmin = new XMLHttpRequest();
xmlhttpAdmin.onreadystatechange = function(){
    if(this.readyState == 4 && this.status == 200) {
        adminTable = new AdminTable(JSON.parse(this.responseText));
    }
};
xmlhttpAdmin.open("GET", "./src/adminList.json", false);
xmlhttpAdmin.send();
```

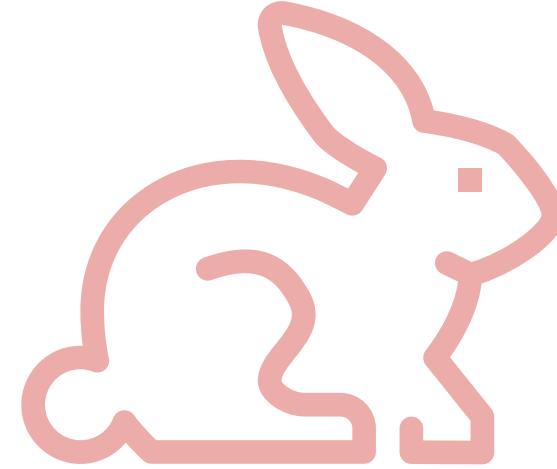
매출 정보를 담은 클래스를 AJAX를 통해 JSON 값으로 INSTANCE 진행



06

END

# Review



이름 : 김푸름

담당 : 홈페이지 HTML, CSS  
작성, 메인 화면 배너,  
레이아웃과 버튼 등 전  
체 디자인 작업

"협업하는 방법을 배울 수 있었습니다."



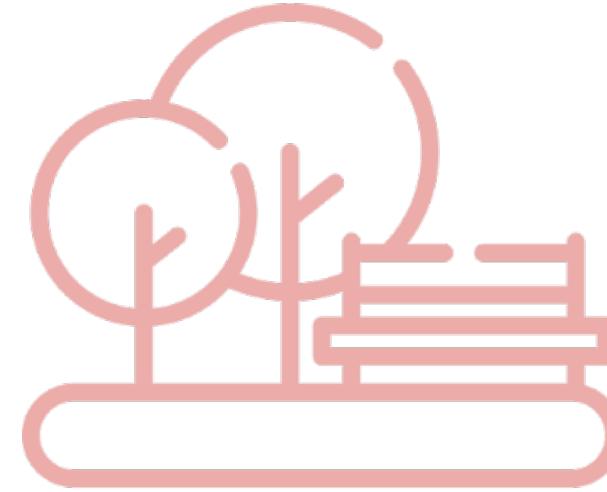
개인작업 위주로 진행을 해오다가 여럿이 하는 프로젝트를 진행하게 되어 걱정이 많았습니다. 한정된 기간 안에 여러 사람이 함께해야 하는 만큼 탭 간격, 변수명부터 주석다는 법, 폰트 사이즈, 버튼 크기 등 프로젝트의 세부적인 것들을 꼼꼼하게 정리하며 작업했습니다. 개인 작업 시에 주먹구구식으로 문제가 생기면 해결했던 것들을 사전에 준비해놓으니 예상보다 더 빠르게 작업할 수 있었습니다.

그리고 다른 사람들이 작업하는 방식들을 보고 배울 수 있는 좋은 기회가 되었습니다.

개인적으로 여러 페이지를 디자인하는 것이 처음이었는데 일관된 컨셉과 방향성을 가지고 디자인하는 것이 생각보다 훨씬 어려운 일이라는 생각이 들었습니다. 연속되어 나오는 페이지이기 때문에 버튼의 크기나 폰트의 크기가 통일되어야 해서 디자인시스템을 꼼꼼히 만들고자 했는데 그럼에도 조금씩 달라져서 힘들었습니다.

팀원들이 응원해주시고 많은 아이디어와 도움을 주셔서 더 빠르고 좋은 결과물이 나온 것 같아 즐거운 경험이었습니다.

# Review



팀장

이름 : 박경호

담당 : 관리자, 포인트 및 결제 부분의 퍼블리싱, SQ L, NodeJS, AJAX 작업, 전체 페이지 HT ML, CSS, JS 코드 일 원화 작업

## "그 어느 때보다 크게 성장한 순간."

학원에 와서 처음으로 공동 작업을 하게 되었습니다. 공동 작업을 하기 전에는 사람이 많으면 당연히 더 좋은 결과물이 빨리 나올 것이라 생각했지만, 막상 프로젝트를 시작하며 팀원들과 이야기를 나누면서 어떻게 작업을 해야 할지 막막하기도 하였습니다. 그래서 더욱이 팀원 다 같이 학원 수업 시작하기 1시간 30분 전, 수업 끝나고 곧바로 사소한 코드 습관부터 로직 구현 방식, 디자인 방향 등등 많은 이야기를 나누었습니다.

팀원들과 계속 소통하며 작업을 분배하고, 서로 작업한 부분에 대해 수정 및 보완할 부분을 피드백해 주다 보니 기존 일정보다 더욱이 앞당겨 작업을 마무리할 수 있었습니다. 팀원 5명 모두 서로의 의견을 존중해 주며 작업을 나아가 아무 충돌 없이, 단 0.1%의 아쉬움 없이 공동 작업을 마무리하고 다같이 결과물을 보니 매우 뿌듯하였습니다.

공동 작업하는 내내 서로에게 배울 점이 정말 많아 혼자 자습하던 때보다 몇 배는 더 효율적으로 공부가 되었고 팀원들의 작업물을 일원화하여 로직에 대한 이해력이 높아졌습니다.

이번 기회를 통해 추후 협업 또는 대형 프로젝트를 진행할 때는 더욱 수월히 작업할 수 있을 것 같습니다. 약 10일간의 작업하는 기간 동안 잠도 얼마 못 자가며 수업, 작업을 이어가느라 다들 고생하였습니다.

# Review



이름 : 박준영

담당 : 상품 선택 화면, 맛 선택 화면, 포인트 옵션 선택 화면 HTML, CSS 및 JavaScript 작성

"일 중독 5명이 뭉쳤다! 함께할 수 있어서 영광이었습니다."



처음에는 각자의 역량, 작업 방식과 구현방식이 다르니 무엇을 어떻게 해야 할지 걱정되고 난감했습니다. 하지만 팀 결성 첫날부터 하루에도 몇 번씩 회의를 통하여 개발 규칙을 만들어 팀만의 독자적인 시스템으로 프로젝트를 진행하여 이러한 걱정은 바로 사라졌습니다.

"변수명을 만들 때 규칙을 지어서 겹치지 않게 하자!", "피그마를 사용해서 작업상황을 공유하자!" 실질적인 프로그래밍에 들어가기 전부터 규칙들이 하나둘 확립되면서 내가 지켜야 하는 시스템이 명확해지니 원활하게 작업에 임할 수 있었습니다.

뿐만 아니라 이번 프로젝트에서 정확하게 역할을 분담하여 자신이 맡은 영역은 끝까지 스스로 책임져서 작업했기 때문에 문제가 발생하거나 수정사항이 필요할 때 높은 코드 이해도로 팀원 간의 빠른 피드백 반영이 이루어져 작업 일정도 많이 단축되었습니다.

이번 프로젝트를 통해 협업이란 무엇인지 경험할 수 있었으며 키오스크를 만들면서 발생했던 여러 버그나 조원들의 피드백을 반영하면서 코드를 수정하는 작업 등 여러 상황을 겪으면서 프로그래밍 능력도 상향되었지만, 팀워크에 대한 능력도 기를 수 있었습니다. 함께한 조원들에게 고마웠고 덕분에 좋은 결과를 얻을 수 있었습니다. 감사합니다!

# Review



이름 : 백지수

담당 : 장바구니 부분 HTML,  
CSS 및 JavaScript  
작성, 사용 설명서와 P  
PT 문서 작업

"함께라서 많은 것을 배울 수 있었습니다."



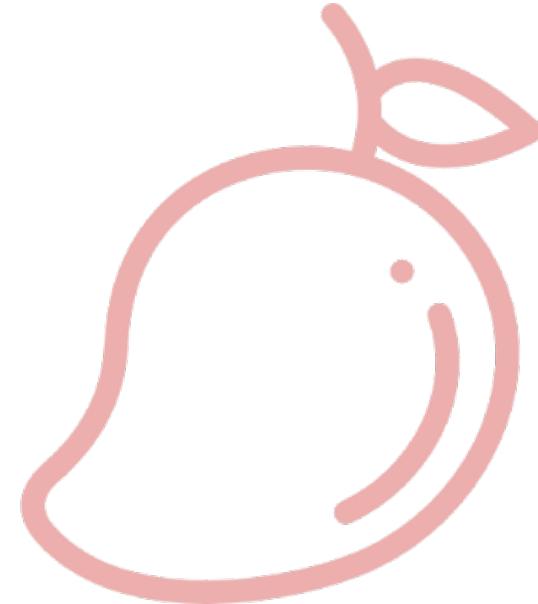
10월 무렵 팀 프로젝트를 시작하게 됐을 때, 처음으로 맡은 공동 작업이라는 생각에 설레기도 했지만 그만큼 걱정이 앞섰습니다. 하지만 여러 차례의 회의를 통해 팀원 사이의 의견을 조율하고, 사용되는 변수 명이나 주석을 적는 방법, HTML의 구조를 짜는 방법, 진행 일정 등 세부적인 부분들을 사전에 정하고 작업에 임하니 혼란스럽지 않게 일을 진행할 수 있었습니다.

팀 프로젝트를 진행하며 가장 어려웠던 점은 서로 주고 받는 데이터의 형식을 정의하고 적절한 형식으로 파싱해서 사용하는 부분이었습니다. 하지만 그 때마다 친절하게 설명해주신 팀원 분들 덕분에 문제를 해결할 수 있었습니다.

또한 디자인을 맡아주신 푸름 님께서 레이아웃부터 CSS, 사용해야 하는 색상과 디자인 요소들을 상세하게 적어주신 덕에 퍼블리싱 작업을 수월하게 해냈습니다. 이 자리를 빌려 다시 한번 감사드립니다!

주도적으로 의견을 내며 문제를 해결하고자 노력하는, 일 중독 네 분과 함께 작업할 수 있어서 정말 영광이었고 스스로에게 느꼈던 아쉬운 점이 있었던 만큼 다음 작업부터는 해당 부분을 보완하여 앞으로 나아가는 사람이 되고 싶습니다. 다들 정말 감사했습니다!

# Review



**이름 :** 최은지

**담당 :** 옵션 페이지 HTML,  
CSS, JavaScript  
작업, 화면 설계서 작성

"열정 넘치는 팀과 함께해서 좋았습니다."



이번 팀 작업을 통해 성취와 배움, 두 마리의 토끼를 잡을 수 있었습니다. 팀 작업은 시작부터 수월했습니다. 서로의 경험을 나누어 초반의 공동 규칙을 세우고, 피그마를 통해 활발히 의견을 교환했습니다. 특히 다른 팀원의 작업 방식을 확인하는 것은 제게 또 다른 학습이 되었습니다.

HTML 태그에 클래스를 주어 CSS를 작성했던 제 작업 방식은 시간이 오래 걸리는 방식이었습니다. 그런데 팀 작업을 통해 배운 선택자를 이용한 방식 덕분에 이전보다 CSS 작업 시간을 절반 가까이 줄였습니다. 또한 빠른 시간 내에 4개의 페이지를 완성한 경험은 저도 상상하지 못했던 큰 성취였습니다. 이 경험을 통해 더 효율적으로 코드 짜는 방법을 배울 수 있었습니다.

그러나 좋은 점이 있다면 아쉬운 점도 있었습니다. 팀원들이 모두 열정적이어서 잠까지 줄여가며 작업을 진행한 탓에 작업 기간 중에는 모두 피로가 역력했습니다. 작업을 기한 내에 끝내는 것도 중요하지만, 건강 또한 중요한 일입니다.

그런데도, 단기간 안에 이런 좋은 결과물을 낸 것은 모두 함께 열정으로 달려 준 팀원들 덕분이라고 생각합니다. 이 경험을 장작 삼아 다른 작업 때는 더 큰 열정을 불태워보겠습니다.

감사합니다



# 배스킨라빈스 키오스크 사용 설명서

김푸름, 박경호, 박준영, 백지수, 최은지



# 목차

- |    |           |       |
|----|-----------|-------|
| 01 | 상품 안내     | p. 3  |
| 02 | 옵션 선택     | p. 4  |
| 03 | 맛 선택      | p. 5  |
| 04 | 장바구니      | p. 6  |
| 05 | 결제하기      | p. 7  |
| 06 | 회원 번호 입력  | p. 8  |
| 07 | 포인트 적립    | p. 9  |
| 08 | 포인트 사용    | p. 10 |
| 09 | 포인트 사용 확인 | p. 11 |
| 10 | 카드 결제     | p. 12 |
| 11 | 결제 완료     | p. 13 |
| 12 | 안내 팝업     | p. 14 |
| 13 | 관리자 페이지   | p. 15 |

# 01 상품 안내

This diagram illustrates the user interface and key features of a mobile POS (mPOS) application, specifically focusing on product selection and payment processing.

- 1 관리자 버튼을 클릭하면 일자별 매출액, 월별 매출액 등을 확인할 수 있는 관리자 모드로 이동합니다.**
- 2 홈 버튼을 클릭하면 모든 내역을 초기화하고 홈 화면으로 이동합니다.**
- 3 탭 메뉴를 누르면 해당 제품의 목록이 화면에 나타납니다.**
- 4 제품 카드를 누르면 해당하는 상품에 맞는 옵션을 선택할 수 있는 화면으로 이동합니다.**
- 5 제품의 개수가 12개를 초과할 경우 우측과 좌측의 화살표 버튼이 활성화되고, 누르면 다른 상품 목록이 나타납니다.**
- 6 구매하고 싶은 모든 제품을 선택하셨다면 '결제하기' 버튼을 눌러주세요.**
- 7 장바구니에 담긴 제품 수가 우측 상단에 나타나고, 그 하단 영역에 제품의 목록 및 옵션이 표시됩니다.**

**Product Categories:** 아이스크림 (Ice Cream), 커피 (Coffee), 케이크 (Cakes)

**Product Options:**

카테고리	제품명	옵션	가격
아이스크림	콜드브루 오트라떼		₩ 4,800
	카페31 아메리카노		₩ 4,500
커피	아메리카노	W 2,800	₩ 2,800
	아이스 아메리카노	W 2,800	₩ 2,800
케이크	아메리카노 (L)	W 3,500	₩ 3,500
	아이스 아메리카노(L)	W 3,500	₩ 3,500
아메리카노	바닐라빈 라떼	W 3,800	₩ 3,800
	아이스 바닐라빈 라떼	W 3,800	₩ 3,800
연유 라떼			₩ 3,800
	아이스 연유 라떼	W 3,800	₩ 3,800
카페 라떼			₩ 3,300
	아이스 카페 라떼	W 3,300	₩ 3,300

## 02 옵션 선택

옵션 선택 → 담기

리얼 27 큐브 ₩ 28,000  
  
9가지 맛 아이스크림 바이트를 초콜릿 판 위에 쌓아올려 전체 다 취식 가능한 실속있는 케이크

**1** 포장  포장을 원한다면 포장하기를 누르고 '+' 버튼으로 시간을 조절해주세요.

포장 안함

아이스 포장 (\*최대 60분)  
- 10 +

보냉백 포장 (\*최대 90분)  
- 10 +

**2** 스푼  원하는 만큼 스푼을 추가할 수 있습니다.

스푼 없음

스푼 추가 (일회용)  
- 1 +

**3** 초  옵션을 전부 선택했다면 '담기' 버튼을 눌러주세요.

큰 초 (\*최대 15개)  
- 0 +

작은 초 (\*최대 15개)  
- 0 +

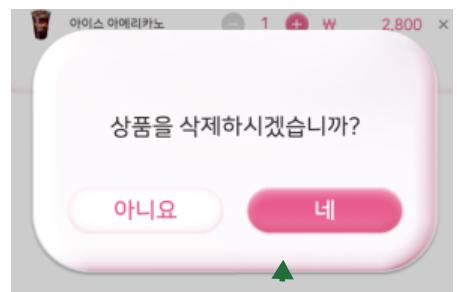
< 이전 화면 → 담기 >

리얼 27 큐브  
  
체리쥬빌레, 뉴욕 치즈케이크, 레인보우 샤베트, 슈팅스타, 민트 ...  
9가지 맛 아이스크림 바이트를 초콜릿 판 위에 쌓아올려 전체 다 취식 가능한 실속있는 케이크

## 03 맛 선택



# 04 장바구니



1단계

주문내역을 확인해 주세요.

1

'+', '-' 버튼으로 제품 수량을 조절할 수 있습니다.

싱글레귤러  
(콘 / 컵)

1



₩

3,500



아메리카노



1



₩

2,800



아이스 카페 라떼



1



₩

3,300



특별한 너를 위한 케이크



1



₩

32,000



뉴욕 치즈케이크, 포장: 포장 안함 | 스푼: 스푼 없음 | 초: 큰 초0개 | 작은 초0…

3

선택 제품이 5개 이상일 때 스크롤을 유도하는 화살표 버튼이 나타납니다. 누르면 가장 하단의 제품으로 스크롤됩니다.

2단계

결제 금액을 확인해 주세요.

₩

46,400

결제하기

4

모든 수량을 결정하셨다면 '결제하기' 버튼을 눌러주세요.

# 05 결제하기

포인트 적립

결제하기

## 할부 선택

적립금이 있으신가요?

1

할부 옵션을 할부 안  
함, 2개월 할부, 5개  
월 할부 중 선택하실  
수 있습니다.

할부 안함

2개월 할부

5개월 할부

## 포인트 사용

적립금이 있으신가요?

2

포인트 적립만 원하  
실 경우 '적립하기'  
버튼을 눌러주세요.  
활성화된 옵션을 한  
번 더 누르면 선택 해  
제됩니다.

적립하기

사용하기

3

포인트 사용과 적립  
모두 원하실 경우 '사  
용하기' 버튼을 눌러  
주세요. 활성화된 옵  
션을 한번 더 누르면  
선택 해제됩니다.

- 포인트 사용 시 자동 적립됩니다.

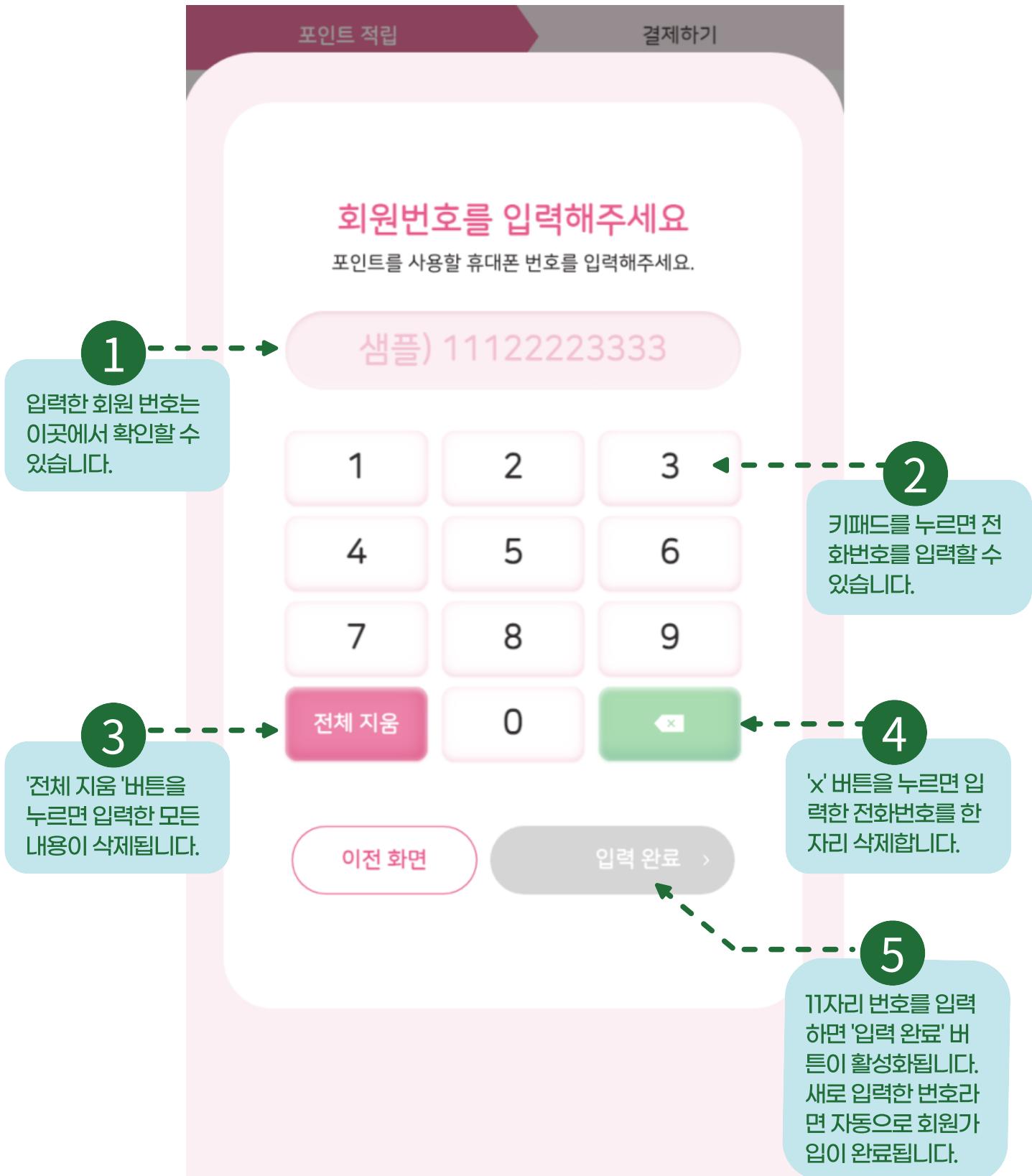
< 이전 화면

다음 단계 (결제하기) >

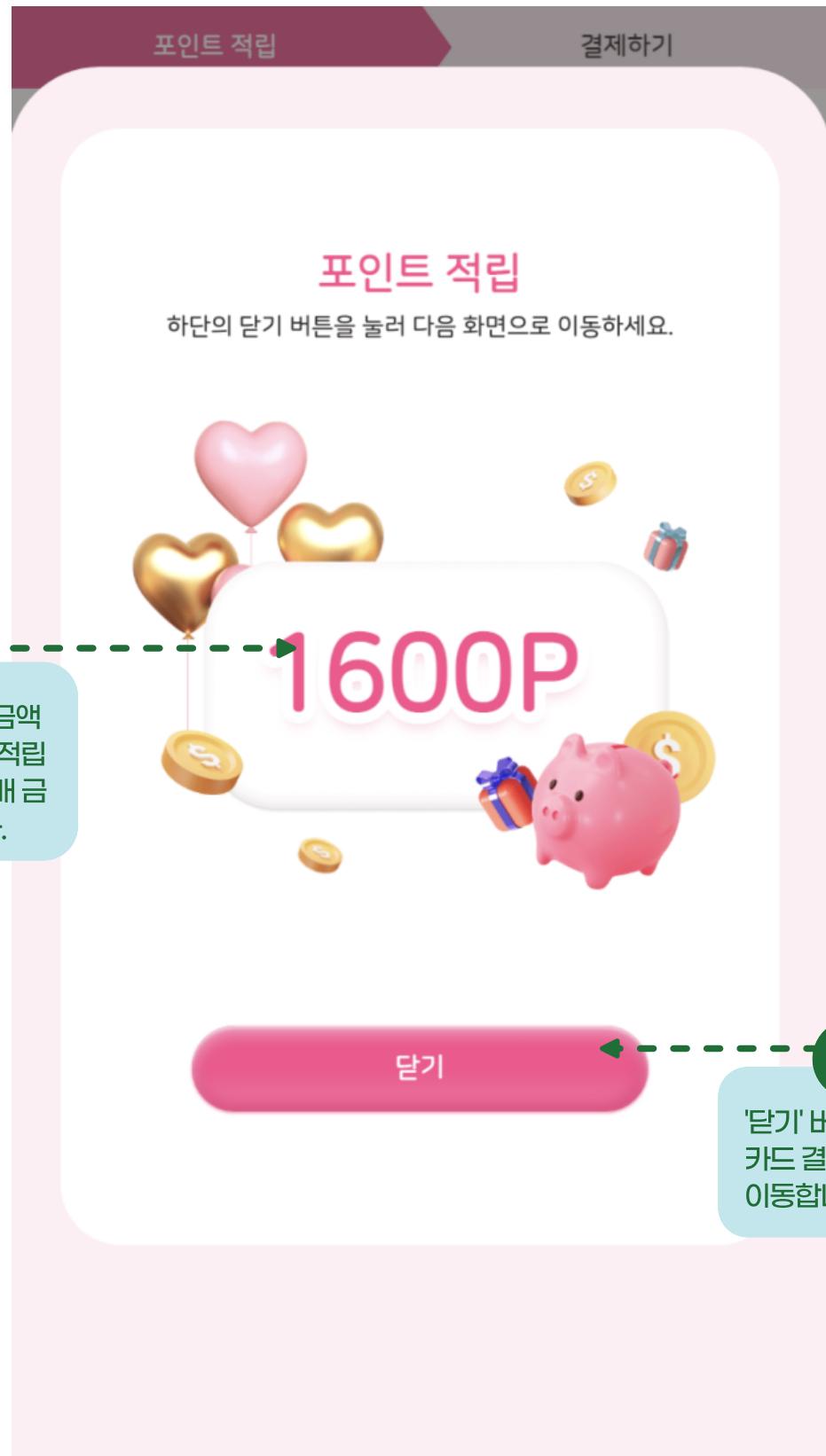
4

결제 옵션을 선택하  
셨다면 '다음 단계'  
버튼을 눌러주세요.

# 06 회원 번호 입력



## 07 포인트 적립



# 08 포인트 사용



# 09 포인트 사용 확인



# 10 카드 결제



# 11 결제 완료



# 12 안내 팝업



# 13 관리자 페이지 - 1

The screenshot shows the BR Manager app interface. At the top, there are three tabs: '일 매출' (Daily Sales) in pink, '월 매출' (Monthly Sales) in light blue, and '매출 순위' (Sales Ranking). A green callout bubble labeled '1' points to the '일 매출' tab, with the text: '탭을 눌러 당일 매출, 해당 월 매출을 확인 할 수 있습니다.' (Tap to check daily sales, monthly sales). Below the tabs, it says '25일' (October 25) and '2022년 10월' (October 2022). To the right is a pink cash register icon with a speech bubble containing three dots.

A green callout bubble labeled '2' points to the '매출액' (Sales Amount) section, which displays '₩ 1,171,600원'. To the right is the '판매 수량' (Sales Quantity) section, which displays '60개'. Below these are three category buttons: '아이스크림' (Ice Cream), '커피' (Coffee), and '케이크' (Cake).

A green callout bubble labeled '3' points to the '케이크' button, with the text: '메뉴 버튼을 누르면 해당하는 정보를 확인 할 수 있습니다.' (Tap the menu button to check the corresponding information).

A green callout bubble labeled '4' points to the bottom section, which is a table titled '제품명' (Product Name) and '수량' (Quantity). The table lists various items and their counts:

제품명	수량
하프갤런	21
버라이어티팩	13
퀵터	8
더블레귤러	5
패밀리	4
싱글킹	4
싱글레귤러	3
더블주니어	2

# 13 관리자 페이지 - 2



일 매출

월 매출

매출 순위

## 2022년



1

순위 탭에서는 카테고리별 판매 금액이 가장 높은 제품을 확인할 수 있습니다.

아이스크림

커피

케이크

순위	제품명	수량	가격
1위	아이스 아메리카노	78	₩ 218,400
2위	아이스 민트 초코 모카	24	₩ 91,200
3위	아이스 연유 라떼	9	₩ 34,200
4위	민트 초코 모카	7	₩ 26,600
5위	아이스 카페 라떼	7	₩ 23,100
6위	아이스 바닐라빈 라떼	6	₩ 22,800
7위	아메리카노	7	₩ 19,600
8위	카페 라떼	5	₩ 16,500
9위	아이스 아메리카노(L)	3	₩ 10,500
10위	카페31 아메리카노	2	₩ 9,000
11위	바닐라빈 라떼	2	₩ 7,600
12위	캬라멜 마끼아또	2	₩ 7,600
13위	콜드브루 오트라떼	1	₩ 4,800

감사합니다!



# 화면 설계서

---

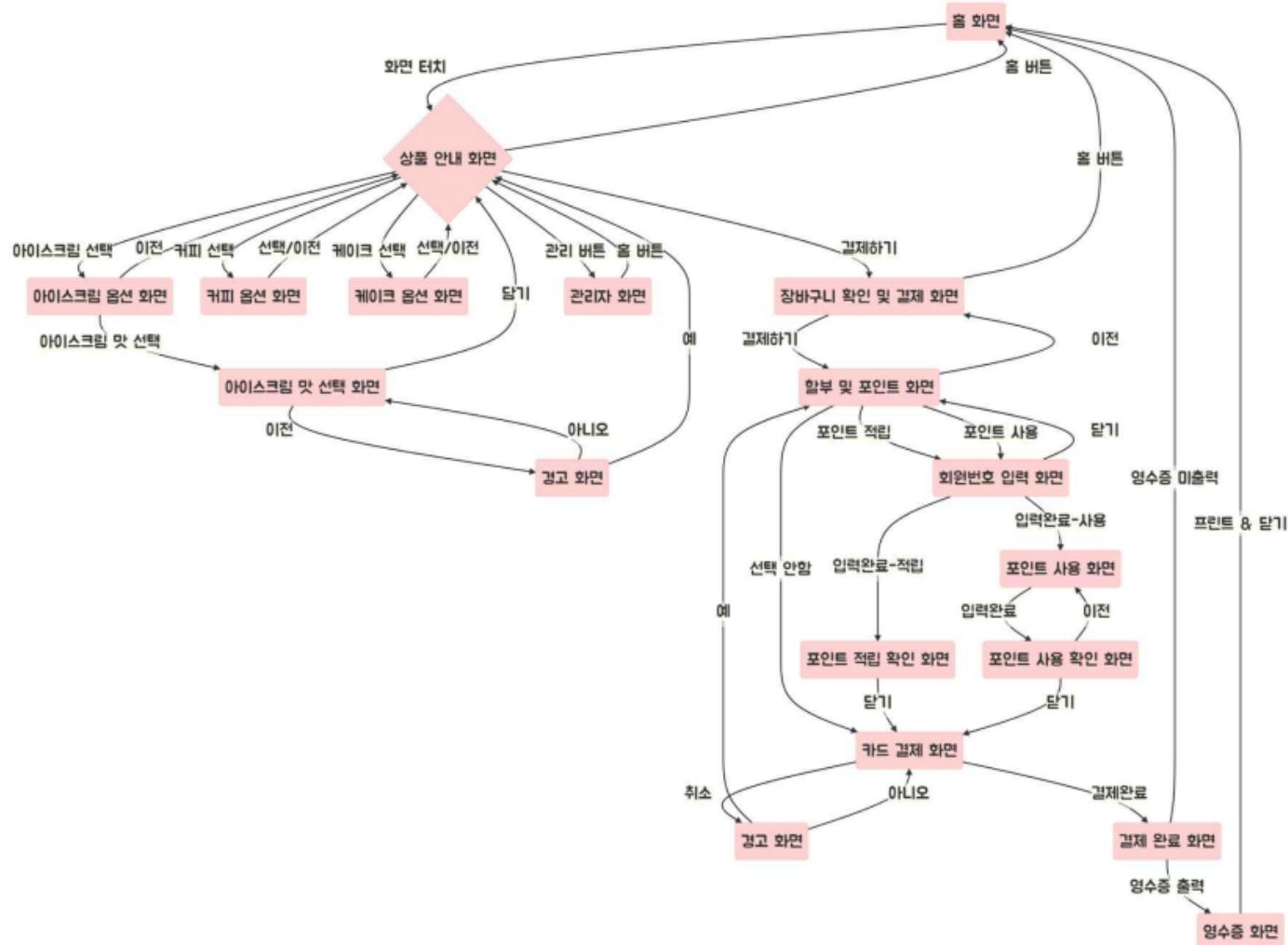
김푸름, 박경호, 박준영, 백지수, 최은지

## History

1. Diagram

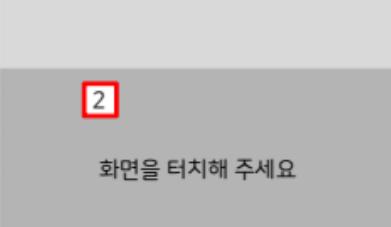
2. 기능 명세서

## 1. Diagram



## 2. 기능 명세서

## 2. 기능 명세서

홈	순번	기능명세서	변수
	1	<b>메인 화면 설정</b> <ul style="list-style-type: none"><li>5장의 아이스크림 컨셉 이미지 슬라이드</li></ul>	
	2	<b>화면 터치 유도</b> <ul style="list-style-type: none"><li>화면 터치를 유도하는 문구 삽입</li></ul>	

## 2. 기능 명세서

상품 안내(1)	순번	기능명세서	변수
	1	<b>관리자 화면</b> <ul style="list-style-type: none"><li>“관리자” 화면으로 전환</li></ul>	<code>timeLogout =&gt; int(setTimeout으로 1초에 1씩 감소)</code> : 화면 유지시간을 나타내는 변수
	2	<b>돌아가기</b> <ul style="list-style-type: none"><li>“홈” 화면으로 돌아감</li></ul>	<code>jsonMain =&gt; object</code> : 전체 json 데이터를 담을 변수
	3	<b>메뉴 탭</b> <ul style="list-style-type: none"><li>각 품목을 누르면 해당하는 상품이 화면에 표시</li><li>상품 클릭 시 해당 영역의 배경색 변화</li></ul>	<code>keyLanguage =&gt; 0(한국어) 1(영어)</code> : 언어를 구분하는 변수
	4	<b>메뉴 전환 버튼</b> <ul style="list-style-type: none"><li>메뉴가 12개를 초과할 경우 해당 버튼 활성화</li><li>버튼 클릭 시, 슬라이드 모션과 함께 화면 전환</li></ul>	<code>keyMainMenu =&gt; 0(아이스크림) 1(커피) 2(케이크)</code> : 탭을 가리키는 변수
	5	<b>선택한 상품에 따른 옵션 화면으로 전환</b> <ul style="list-style-type: none"><li>아이스크림일 경우 “아이스크림 옵션” 화면으로 전환</li><li>커피일 경우 “커피 옵션” 화면으로 전환</li><li>케이크일 경우 “케이크 옵션” 화면으로 전환</li></ul>	<code>countPage =&gt; int</code> : 페이지네이션 번호 변수

## 2. 기능 명세서

상품 안내(2)	순번	기능명세서	변수
	6	<p><b>장바구니에 담긴 상품의 개수와 정보 표시</b></p> <ul style="list-style-type: none"> <li>‘장바구니’ 버튼 오른쪽 상단에 담긴 상품의 개수 표시</li> <li>담은 물건이 ‘결제하기’ 버튼 하단에 가로로 나열</li> <li>만약 장바구니 내 상품이 3개를 초과할 경우 좌우로 이동할 수 있는 버튼 생성</li> </ul>	<b>arrMainCart =&gt; arr [object]</b> : 장바구니 배열 [제이슨 객체]  <b>countMainCart =&gt; arr [int]</b> : 장바구니 배열에 담긴 제이슨 객체의 각각의 수량이 담기는 배열
	7	<p><b>결제하기 버튼을 누르면 결제 화면으로 전환</b></p> <ul style="list-style-type: none"> <li>장바구니에 담긴 것이 없을 경우 ‘결제하기’ 버튼 비활성화</li> <li>상품이 하나 이상 담긴 경우 결제창의 배경색이 바뀌며 ‘결제하기’ 버튼 활성화</li> <li>‘결제하기’ 버튼 클릭 시 “장바구니 결제” 화면으로 이동</li> </ul>	<b>countSumMainCart =&gt; int</b> : 7번 위치에 표시되는 변수로서 장바구니 메뉴 총합  <b>priceSumPay =&gt; int</b> : 결제하기 버튼에 표시되는 금액으로서 장바구니 메뉴들의 총액
	8	<p><b>장바구니에 담은 물건 삭제</b></p> <ul style="list-style-type: none"> <li>담은 물건 우측 상단의 ‘X’ 표시 버튼을 누르면 해당 제품을 목록에서 삭제</li> </ul>	

## 2. 기능 명세서

아이스크림 옵션	순번	기능명세서	변수
	1	<p><b>포장 옵션</b></p> <ul style="list-style-type: none"> <li>각 옵션 클릭 시 보더의 색이 변함</li> <li>기본 시간은 10분이며 '+' 버튼 클릭 시 5분 단위로 시간 추가, '-' 버튼 클릭 시 5분 단위로 시간이 감소</li> <li>최솟값은 0이며 아이스 포장은 60분, 보냉백 포장은 90분이 최대</li> <li>최소 시간에 도달한 경우 '-' 버튼 비활성화, 최대 시간에 도달한 경우 '+' 버튼 비활성화</li> </ul>	<b>priceSelectMenu =&gt; int</b> : 선택 메뉴의 가격을 담을 변수
	2	<p><b>스푼 옵션</b></p> <ul style="list-style-type: none"> <li>각 옵션 클릭 시 보더의 색이 변함</li> <li>선택한 상품에 맞는 스푼 개수를 기본적으로 설정 (ex. 싱글 1개, 쿼터 2개 등)</li> <li>'+'버튼을 누를 때마다 1씩 추가, '-'버튼을 누르면 1씩 감소 (추가 요금 없음)</li> <li>최솟값은 1이며 최댓값은 99</li> <li>최솟값에 도달한 경우 '-' 버튼 비활성화, 최댓값에 도달한 경우 '+' 버튼 비활성화</li> </ul>	

## 2. 기능 명세서

아이스크림 옵션	순번	기능명세서	변수
	3	<b>이전 화면 버튼</b> <ul style="list-style-type: none"> <li>클릭 시 이전 화면인 “상품 안내” 화면으로 돌아감</li> </ul>	
	4	<b>맛 선택 화면 전환</b> <ul style="list-style-type: none"> <li>‘맛 선택’ 버튼 클릭 시 다음 단계인 “맛 선택” 화면으로 전환</li> </ul>	
	5	<b>선택 상품 설명</b> <ul style="list-style-type: none"> <li>선택한 상품의 이름과 설명, 사진 표시</li> </ul>	

## 2. 기능 명세서

커피 옵션	순번	기능명세서	변수
 <p>1 2 3</p>	1	<b>상품 정보 안내</b> <ul style="list-style-type: none"> <li>해당 상품의 이미지, 이름, 설명, 가격을 띠움</li> </ul>	
	2	<b>포장하기</b> <ul style="list-style-type: none"> <li>커피는 아이스, 보냉백 포장이 필요없으므로 포장 안함과 포장으로 나눔.</li> </ul>	
	3	<b>샷 추가, 시럽 추가</b> <ul style="list-style-type: none"> <li>각 옵션 클릭 시 보더의 색이 변함</li> <li>'+'버튼을 누를 때마다 1씩 추가, '-'버튼을 누르면 1씩 감소 (추가 요금 없음)</li> <li>최솟값은 1이며, 최댓값은 10</li> <li>최솟값에 도달한 경우 '-' 버튼 비활성화, 최댓값에 도달한 경우 '+' 버튼 비활성화</li> </ul>	

## 2. 기능 명세서

커피 옵션	순번	기능명세서	변수
	4	<b>이전 화면 버튼</b> <ul style="list-style-type: none"> <li>클릭 시 이전 화면인 “상품 안내” 화면으로 이동</li> <li>이때 “상품 안내” 화면의 하단에 해당 상품이 담겨있지 않음</li> </ul>	
	5	<b>담기 버튼</b> <ul style="list-style-type: none"> <li>클릭 시 “상품 안내” 화면으로 이동</li> <li>이때 “상품 안내” 화면의 하단에 해당 상품이 담김</li> </ul>	
	6	<b>상품 설명</b> <ul style="list-style-type: none"> <li>상품의 이미지, 이름, 설명글과 칼로리, 알레르기 성분 등이 있음</li> </ul>	

## 2. 기능 명세서

케이크 옵션(1)	순번	기능명세서	변수
 <p>1. 옵션 선택 → 2. 맛 선택</p> <p>1. 헤피 버스데이 케이크 ₩ 26,000 리얼로 감성의 그림 이미지와 화려한 색상, 레이어링 초콜릿 장식으로 모두의 창업을 특별하게 만들어 줄 케이크</p> <p>2. 포장 포장 단장 (선택) 아이스 포장 (*최대 10분) 보냉백 포장 (*최대 20분) 80 90</p> <p>3. 스푼 스푼 단장 스푼 후기 (일회용) 99</p> <p>4. 쿠션 쿠션 단장 (*최대 10개) 작은 쿠션 (*최대 20개) 10 10</p> <p>&lt; 이전 화면 담기 &gt;</p> <p>헤피 버스데이 케이크 할로윈 강보   할로윈 테마를 발생시킬 수 있는 내용내용내용내용 케이크로 감성의 그림 이미지와 화려한 색상, 레이어링 초콜릿 장식으로 모두의 창업을 특별하게 만들어 줄 케이크</p>	1	<b>상품 정보 안내</b> <ul style="list-style-type: none"> <li>해당 상품의 이미지, 이름, 설명, 가격을 띠움</li> </ul>	
	2	<b>포장 옵션</b> <ul style="list-style-type: none"> <li>각 옵션 클릭 시 보더의 색이 변함</li> <li>기본 시간은 10분이며 '+' 버튼 클릭 시 5분 단위로 시간 추가, '-' 버튼 클릭 시 5분 단위로 시간이 감소</li> <li>최솟값은 0이며 드라이아이스 포장은 60분, 보냉백 포장은 90분이 최대</li> <li>최소 시간에 도달한 경우 '-' 버튼 비활성화, 최대 시간에 도달한 경우 '+' 버튼 비활성화</li> </ul>	
	3	<b>스푼 옵션</b> <ul style="list-style-type: none"> <li>각 옵션 클릭 시 보더의 색이 변함</li> <li>기본값은 '1'이며, '+'버튼을 누를 때마다 1씩 추가, '-'버튼을 누르면 1씩 감소(추가 요금 없음)</li> <li>최솟값은 0, 최댓값은 99</li> <li>최솟값에 도달한 경우 '-' 버튼 비활성화, 최댓값에 도달한 경우 '+' 버튼 비활성화</li> </ul>	

## 2. 기능 명세서

케이크 옵션(2)	순번	기능명세서	변수
 <p>4</p>	4	<b>초 옵션</b> <ul style="list-style-type: none"> <li>각 옵션 클릭 시 보더의 색이 변함</li> <li>기본값은 0이며, '+'버튼을 누를 때마다 1씩 추가, '-'버튼을 누르면 1씩 감소(추가 요금 없음)</li> <li>최솟값은 0, 최댓값은 99</li> <li>최솟값에 도달한 경우 '-' 버튼 비활성화, 최댓값에 도달한 경우 '+' 버튼 비활성화</li> </ul>	
<p>5</p>	5	<b>이전 화면 버튼</b> <ul style="list-style-type: none"> <li>클릭 시 이전 화면인 “상품 안내” 화면으로 이동</li> <li>이때 “상품 안내” 화면의 하단에 해당 상품이 담겨있지 않음</li> </ul>	
<p>6</p>	6	<b>담기 버튼</b> <ul style="list-style-type: none"> <li>클릭 시 “상품 안내” 화면으로 이동</li> <li>이때 “상품 안내” 화면의 하단에 해당 상품이 담김</li> </ul>	
<p>7</p>	7	<b>상품 설명</b> <ul style="list-style-type: none"> <li>상품의 이미지, 이름, 설명글과 칼로리, 알레르기 성분 등이 있음</li> </ul>	

## 2. 기능 명세서

콘/컵 옵션	순번	기능명세서	변수
 <p>1 상글 레귤러 (콘 / 컵) W 3,500 할가지 모음 쓰스힐게 줄줄 수 있는 시리즈</p> <p>2 컨 / 컵</p> <p>3 스푼</p> <p>4 이전 화면</p> <p>5 담기</p> <p>6 상글 레귤러 (콘 / 컵) 상글과 함께   알약포기자를 발생시킬 수 있는 내용내용내용내용 본 제품을 쓰시 즐기는 제품으로 드려야 서비스 조장이 되도록 합니다.</p>	1	<b>상품 정보 안내</b> <ul style="list-style-type: none"> <li>해당 상품의 이미지, 이름, 설명, 가격을 띠움</li> </ul>	
	2	<b>콘/컵 옵션</b> <ul style="list-style-type: none"> <li>각 옵션 클릭 시 보더의 색이 변함</li> </ul>	
	3	<b>스푼 옵션</b> <ul style="list-style-type: none"> <li>각 옵션 클릭 시 보더의 색이 변함</li> <li>‘+’버튼을 누를 때마다 1씩 추가, ‘-’버튼을 누르면 1씩 감소 (추가 요금 없음)</li> <li>최솟값은 1이며 최댓값은 99</li> <li>최솟값에 도달한 경우 ‘-’ 버튼 비활성화, 최댓값에 도달한 경우 ‘+’ 버튼 비활성화</li> </ul>	
	4	<b>이전 화면 버튼</b> <ul style="list-style-type: none"> <li>클릭 시 “상품 안내” 화면으로 이동</li> <li>이때 “상품 안내” 화면의 하단에 해당 상품이 담기지 않음</li> </ul>	
	5	<b>담기 버튼</b> <ul style="list-style-type: none"> <li>클릭 시 “상품 안내” 화면으로 이동</li> <li>이때 “상품 안내” 화면의 하단에 해당 상품이 담김</li> </ul>	
	6	<b>상품 설명</b> <ul style="list-style-type: none"> <li>상품의 이미지, 이름, 설명글이 있음</li> </ul>	

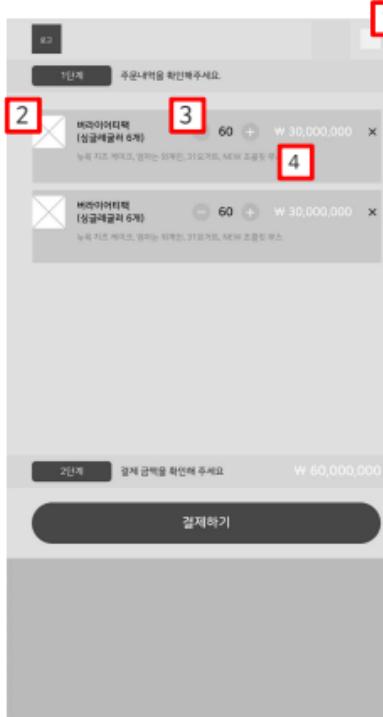
## 2. 기능 명세서

맛 선택(1)	순번	기능명세서	변수
	1	<b>현재 화면 설정</b> <ul style="list-style-type: none"> <li>맛이 12개 이상일 경우 하단에 페이지네이션 및 오른쪽에 화살표 생성</li> <li>신제품 → 인기도 → 랜덤으로 정렬             <ul style="list-style-type: none"> <li>신제품일 경우 이미지 위에 'New' 이미지 표시</li> <li>인기도 1위, 2위, 3위일 경우 이미지 오른쪽 상단에 왕관 마크와 함께 해당 순위 표시</li> </ul> </li> </ul>	<code>arrSelectFlavor =&gt; [-1, -1, -1, -1, -1, -1]</code> 고정 : 맛 Num값 담을 배열
	2	<b>선택한 맛이 아래로 이동</b> <ul style="list-style-type: none"> <li>맛 클릭 시 이미지가 opacity, scale 변화와 함께 장바구니로 이동하는 모션 적용</li> </ul>	
	3	<b>맛 페이지 이동</b> <ul style="list-style-type: none"> <li>버튼 클릭 시 페이지 이동</li> <li>해당 페이지의 페이지네이션 색깔 바뀜</li> </ul>	
	4	<b>맛 담기</b> <ul style="list-style-type: none"> <li>선택한 상품에 따라 맛의 개수 만큼 미리 빈 공간 표시 (최대 6개)</li> <li>'X' 버튼 클릭 시 해당 맛 삭제</li> <li>삭제 시 삭제된 곳은 빈칸으로 놓아두고, 이후 맛 재선택 시 빈곳에 들어감</li> <li>정해진 개수를 초과해서 담을 시 적용되지 않음</li> </ul>	

## 2. 기능 명세서

맛 선택(2)	순번	기능명세서	변수
  <b>6</b> < 이전 화면 <b>7</b> 담기 >	6	<p><b>이전 화면 버튼</b></p> <ul style="list-style-type: none"> <li>클릭 시 알림창 팝업</li> </ul> <p><b>6-1) '예' 버튼을 클릭할 경우</b></p> <ul style="list-style-type: none"> <li>- 이전 화면인 “아이스크림 옵션” 화면으로 이동</li> </ul> <p><b>6-2) '아니오' 버튼을 클릭할 경우</b></p> <ul style="list-style-type: none"> <li>- 이전 화면으로 이동하지 않음</li> </ul> <p><b>Tooltip</b></p> <ul style="list-style-type: none"> <li>이전 화면으로 이동하시겠습니까? (현재 고르신 상품은 전부 초기화됩니다.)</li> </ul>	
 <b>8</b>	7	<p><b>담기 버튼</b></p> <ul style="list-style-type: none"> <li>정해진 개수만큼 맛 선택 시 ‘담기’ 버튼 활성화</li> <li>클릭 시 “상품 안내” 화면 이동</li> <li>이때 “상품 안내” 화면 하단에 해당 상품이 담김</li> </ul>	
	8	<p><b>맛 설명</b></p> <ul style="list-style-type: none"> <li>맛 선택 시 해당 맛의 이름 및 설명, 칼로리, 알레르기 성분 등이 있음</li> </ul>	

## 2. 기능 명세서

장바구니 결제(1)	순번	기능명세서	변수
 <p>1단계 주문내역을 확인해주세요.</p> <p>2단계 베리아이어티백 [상글에글리 5개] 60 + ₩ 30,000,000 × 눌러 기준, 빼기, 즐기는 외계인, 고요거림, 세비 조류등 퀴즈</p> <p>3단계 베리아이어티백 [상글에글리 6개] 60 + ₩ 30,000,000 × 눌러 기준, 빼기, 즐기는 외계인, 고요거림, 세비 조류등 퀴즈</p> <p>2단계 결제 금액을 확인해 주세요 ₩ 60,000,000</p> <p>결제하기</p>	1	<b>홈 버튼 설정</b> <ul style="list-style-type: none"> <li>‘홈’ 버튼 클릭 시 “홈” 화면으로 돌아감</li> <li>아이콘은 집 모양 형태</li> </ul>	
	2	<b>그림 영역</b> <ul style="list-style-type: none"> <li>상품 선택, 장바구니 등에 있던 상품의 이미지 첨부</li> </ul>	
	3	<b>수량 설정</b> <ul style="list-style-type: none"> <li>구매한 상품의 수량 변경이 가능함</li> <li>3-1) ‘1’인 경우           <ul style="list-style-type: none"> <li>‘-’ 버튼 비활성화</li> </ul> </li> <li>3-2) ‘2’ 이상인 경우           <ul style="list-style-type: none"> <li>‘-’ 버튼 활성화</li> </ul> </li> <li>최솟값은 1, 최댓값은 99</li> <li>최댓값에 도달한 경우 ‘+’ 버튼 비활성화</li> <li>수량을 늘리고 줄임과 동시에 계산된 가격 적용</li> </ul>	
	4	<b>상품 이름 및 상품 설명</b> <ul style="list-style-type: none"> <li>장바구니에 담은 상품의 이름과 설명, 고른 옵션 등이 들어가 있음</li> </ul>	

## 2. 기능 명세서

장바구니 결제(2)	순번	기능명세서	변수
	5	<p><b>장바구니 내 상품 삭제</b></p> <ul style="list-style-type: none"> <li>‘X’ 버튼 클릭 시 알림창 팝업</li> </ul> <p><b>5-1) ‘예’ 버튼을 클릭할 경우</b></p> <ul style="list-style-type: none"> <li>- 상품 삭제</li> </ul> <p><b>5-2) ‘아니오’ 버튼을 클릭할 경우</b></p> <ul style="list-style-type: none"> <li>- 상품을 삭제하지 않음</li> </ul> <p><b>Tooltip</b></p> <ul style="list-style-type: none"> <li>상품을 삭제하시겠습니까?</li> </ul>	
	6	<p><b>최종 금액</b></p> <ul style="list-style-type: none"> <li>계산된 최종 금액을 띠움</li> <li>세 자리 수마다 ‘,’ 표기</li> </ul>	
	7	<p><b>결제하기</b></p> <ul style="list-style-type: none"> <li>‘결제하기’ 버튼 클릭 시 “결제하기” 화면으로 이동</li> </ul>	

## 2. 기능 명세서

결제하기	순번	기능명세서	변수
 <p>포인트 적립</p> <p>할부 선택 적립금이 있으신가요?</p> <p>1 할부 안함      2개월 할부      5개월 할부</p> <p>포인트 사용 적립금이 있으신가요?</p> <p>2 적립하기      3 사용하기</p> <p>▶ 포인트 사용 시 적용 차입됩니다.</p> <p>4 이전 화면      5 다음 단계 (결제하기) &gt;</p>	1	<b>카드 할부 선택</b> <ul style="list-style-type: none"> <li>기본값은 ‘할부 안함’이며, 5만 원 이상 구매 시 나머지 할부 선택 버튼 클릭 가능</li> </ul>	<b>countMonthly</b> => 1 / 2 / 5 정수값 : 영수증에서 할부 출력 필요
	2	<b>적립하기</b> <ul style="list-style-type: none"> <li>‘적립하기’ 버튼 클릭 시 색이 바뀌며 활성화</li> <li>이후 ‘결제하기’ 버튼 클릭 시 “회원 번호 입력” 화면으로 이동</li> </ul>	<b>keySelectionPoint</b> => 0(선택안함) / 1(적립) / 2(사용) : point 사용 유무 키값
	3	<b>사용하기</b> <ul style="list-style-type: none"> <li>‘사용하기’버튼 클릭 시 색이 바뀌며 활성화</li> <li>이 후 ‘결제하기’ 버튼 클릭 시 “회원 번호 입력” 화면으로 이동</li> <li>적립 및 포인트 사용 모두 가능</li> </ul>	
	4	<b>이전 화면 버튼</b> <ul style="list-style-type: none"> <li>클릭 시 이전 화면인 “장바구니 결제” 화면으로 이동</li> </ul>	
	5	<b>결제하기</b> <ul style="list-style-type: none"> <li>‘사용하기’ 혹은 ‘적립하기’ 버튼을 클릭하지 않았을 경우 “카드 결제” 화면으로 이동</li> </ul>	

## 2. 기능 명세서

회원 번호 입력	순번	기능명세서	변수
	1	<b>입력한 회원 번호 표시</b> <ul style="list-style-type: none"> <li>하단의 키패드로 입력하면 입력한 번호 표시</li> <li>번호가 모두 입력될 경우 ‘입력 완료’ 버튼 활성화</li> <li>번호는 11자리를 초과하여 입력할 수 없도록 제한</li> </ul>	<b>inputMemberNum =&gt; string(11)</b> : 사용자에게 입력받은 핸드폰 번호를 담는 변수
	2	<b>번호 입력 키패드</b> <ul style="list-style-type: none"> <li>0~9의 숫자 입력 가능 키패드</li> <li>‘전체 지움’ 버튼 클릭 시 사용 표시에 출력된 값 전체 삭제</li> <li>‘X’ 버튼 클릭 시 출력된 값 한 자리 삭제</li> </ul>	
	3	<b>이전 화면 버튼</b> <ul style="list-style-type: none"> <li>클릭 시 이전 화면인 “결제하기” 화면으로 이동</li> </ul>	
	4	<b>입력 완료 버튼</b> <ul style="list-style-type: none"> <li>클릭 시 입력한 휴대폰 번호를 “회원” 테이블에 있는 번호와 비교, 만약 신규 번호일 시 테이블에 번호와 1000P를 추가</li> </ul> <p>4-1) 적립하기로 넘어온 경우</p> <ul style="list-style-type: none"> <li>“포인트 적립”화면으로 이동</li> </ul> <p>4-2) 사용하기로 넘어온 경우</p> <ul style="list-style-type: none"> <li>“포인트 사용” 화면으로 이동</li> </ul>	

## 2. 기능 명세서

포인트 적립	순번	기능명세서	변수
 <p>포인트 적립 하단의 닫기 버튼을 눌러 다음 화면으로 이동하세요.</p> <p>1 10P</p> <p>2 닫기</p>	1	<b>적립 포인트 표시</b> <ul style="list-style-type: none"><li>적립 포인트 산출 기준에 따른 포인트 적립금 표시</li><li>적립금은 전체 결제 금액의 5%</li><li>해당 화면에 출력된 포인트는 적립 ‘예정금’</li></ul>	pointSave => 적립금 저장 및 출력 (영수증 출력)  pointUse => 사용포인트 input태그 value값 저장 (영수증 출력)  pointMember => DB에서 가져온 회원 포인트 값 (영수증 출력)
	2	<b>닫기 버튼</b> <ul style="list-style-type: none"><li>“포인트 적립” 화면 종료</li><li>“카드 결제” 화면으로 이동</li></ul>	

## 2. 기능 명세서

포인트 사용(1)	순번	기능명세서	변수
	1	<b>보유 포인트 표시</b> <ul style="list-style-type: none"><li>“회원 번호 입력” 화면에서 입력한 휴대폰 번호로 “회원” 테이블에서 보유 포인트 값을 조회하여 표시</li></ul>	
	2	<b>사용 포인트 표시</b> <ul style="list-style-type: none"><li>하단의 키패드를 통해 입력한 포인트 값이 표시</li><li>보유 포인트 값보다 높은 값 입력 불가</li><li>상품 금액을 초과하는 포인트 입력 불가</li><li>1000P 미만의 값을 입력할 경우 ‘입력 완료’ 버튼 비활성화</li></ul>	

## 2. 기능 명세서

포인트 사용(2)	순번	기능명세서	변수												
 <p>보유 포인트 P 사용 포인트 P • 포인트는 1000p마다 사용 가능합니다.</p> <p>3</p> <table border="1"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>4</td><td>5</td><td>6</td></tr> <tr><td>7</td><td>8</td><td>9</td></tr> <tr><td>전체 지움</td><td>0</td><td>■</td></tr> </table> <p>이전 화면 4 입력 완료 5 &gt;</p>	1	2	3	4	5	6	7	8	9	전체 지움	0	■	3	<p><b>숫자 입력 키패드</b></p> <ul style="list-style-type: none"> <li>0~9 숫자 입력 가능 키패드</li> <li>‘전체 지움’ 버튼 클릭 시 사용 표시에 출력된 값 전체 삭제</li> <li>‘X’ 버튼 클릭 시 출력된 값 한 자리 삭제</li> </ul>	
1	2	3													
4	5	6													
7	8	9													
전체 지움	0	■													
	4	<p><b>이전 화면 버튼</b></p> <ul style="list-style-type: none"> <li>클릭 시 이전 화면인 “결제하기” 화면으로 이동</li> </ul>													
	5	<p><b>입력 완료 버튼</b></p> <ul style="list-style-type: none"> <li>결제가 완료될 경우 “회원” 테이블에서 보유한 포인트 값을 사용 포인트만큼 차감</li> <li>단, 현재 화면이 아닌 “결제 완료” 화면에서 사용 포인트 값 만큼을 차감(이 단계에서는 사용 예정인 포인트이기 때문)</li> </ul>													

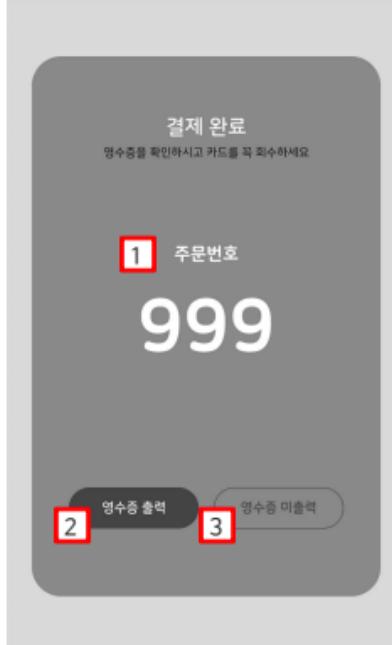
## 2. 기능 명세서

포인트 사용 확인	순번	기능명세서	변수
 <p>1 총 금액 W 30,000,000 포인트 사용량 - 10,000p 적립 포인트 1,499,500p</p> <p>2 최종 결제 금액 W 29,990,000</p> <p>3 이전 화면 &lt;      4 결제하기 &gt;</p>	1	<b>결제 금액 계산</b> <ul style="list-style-type: none"> <li>총 금액은 포인트를 사용하기 전 금액</li> <li>포인트 사용량은 “포인트 사용” 화면에서 입력한 포인트를 출력</li> <li>적립 포인트는 최종 결제 금액의 5%</li> <li>세 자릿 수마다 ‘,’ 표시</li> </ul>	priceLastPay => 포인트 차감한 최종 결제 금액
	2	<b>최종 결제 금액</b> <ul style="list-style-type: none"> <li>(총 금액 - 포인트 사용량)의 값을 최종 결제 금액으로 출력</li> <li>세 자릿 수마다 ‘,’ 표시</li> </ul>	
	3	<b>이전 화면 버튼</b> <ul style="list-style-type: none"> <li>클릭 시 이전 화면인 “포인트 사용” 화면으로 돌아감</li> </ul>	
	4	<b>결제하기 버튼</b> <ul style="list-style-type: none"> <li>클릭 시 “카드 결제” 화면으로 넘어감</li> </ul>	

## 2. 기능 명세서

카드 결제	순번	기능명세서	변수
	1	<b>애니메이션</b> <ul style="list-style-type: none"><li>사용자가 신용카드를 올바르게 넣을 수 있도록 유도하는 애니메이션</li></ul>	
	2	<b>취소 버튼</b> <ul style="list-style-type: none"><li>클릭 시 알림창 팝업<ul style="list-style-type: none"><li>2-1) '예' 버튼을 클릭했을 경우<ul style="list-style-type: none"><li>“결제하기” 화면으로 이동</li></ul></li><li>2-2) '아니오' 버튼을 클릭했을 경우<ul style="list-style-type: none"><li>결제 진행</li></ul></li></ul></li></ul> <p><b>Tooltip</b></p> <ul style="list-style-type: none"><li>주문을 취소하시겠습니까?</li></ul>	
	3	<b>결제 완료 버튼</b> <ul style="list-style-type: none"><li>사용자가 신용카드를 삽입한 뒤 해당 버튼을 클릭하면 결제가 완료되어 다음 화면으로 이동</li></ul>	

## 2. 기능 명세서

결제 완료	순번	기능명세서	변수
 결제 완료 영수증을 확인하시고 카드를 꺼 회수하세요  1 주문번호 999  2 영수증 출력      3 영수증 미출력	1	<b>주문번호</b> <ul style="list-style-type: none"><li>결제일 기준 몇 번째 주문 고객인지 확인 가능</li><li>자동 기준 주문번호는 1번으로 초기화</li><li>소비자가 주문한 음식이 맞는지 확인 가능</li></ul>	countOrder => 주문번호 : 하루마다 1로 초기화 되는거 구현
	2	<b>영수증 미출력 버튼</b> <ul style="list-style-type: none"><li>클릭 시 영수증 없이 주문번호 용지만 출력</li><li>“홈” 화면으로 자동 이동</li></ul>	
	3	<b>영수증 출력 버튼</b> <ul style="list-style-type: none"><li>클릭 시 주문번호를 포함한 영수증이 출력</li><li>“영수증 확인” 화면으로 이동</li></ul>	

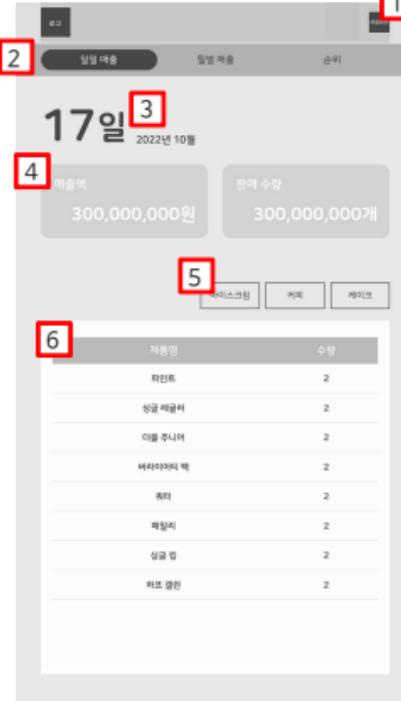
## 2. 기능 명세서

영수증 확인(1)	순번	기능명세서	변수
 <p>The image shows a Baskin Robbins receipt with several points of interest highlighted by red boxes and numbers:</p> <ul style="list-style-type: none"> <li><b>1</b>: 영수증 (Receipt)</li> <li><b>2</b>: PRINT 버튼 (PRINT button)</li> <li><b>3</b>: 닫기 버튼 (Close button)</li> <li><b>4</b>: 주문번호 (Order Number) 999</li> <li><b>5</b>: 안산 고잔점 대표자: 홍길동 경기도 안산시 단원구 (Ansan Goyang Branch Manager: Hong Gil-dong Gyeonggi-do Ansan-si Danwon-gu)</li> <li><b>6</b>: 판매점 정보 (Sales point information) showing items and their details.</li> </ul>	<p><b>영수증</b></p> <ul style="list-style-type: none"> <li>화면 로딩과 동시에 즉시 영수증 출력</li> <li>주문 이력이 많다면 스크롤로 내용 확인 가능</li> </ul> <p><b>PRINT 버튼</b></p> <ul style="list-style-type: none"> <li>'PRINT' 버튼 클릭 시 영수증 출력</li> </ul> <p><b>닫기 버튼</b></p> <ul style="list-style-type: none"> <li>클릭 시 “홈” 화면으로 이동</li> </ul> <p><b>주문번호</b></p> <ul style="list-style-type: none"> <li>결제일 기준 몇 번째 주문 고객인지 확인 가능</li> <li>자동 기준 주문번호는 1번으로 초기화</li> <li>소비자가 주문한 음식이 맞는지 확인 가능</li> </ul> <p><b>판매점 정보</b></p> <ul style="list-style-type: none"> <li>주문을 진행한 가게의 정보</li> </ul> <p><b>주문 내역</b></p> <ul style="list-style-type: none"> <li>소비자가 결제한 제품의 내역</li> </ul>		

## 2. 기능 명세서

영수증 확인(2)	순번	기능명세서	변수
	7	<b>총 결제 금액</b> <ul style="list-style-type: none"> <li>주문 금액의 총합계 표시</li> </ul>	
	8	<b>해피포인트 사용 내역</b> <ul style="list-style-type: none"> <li>소비자가 사용한 해피포인트를 표시</li> </ul>	
	9	<b>신용카드 결제 내역</b> <ul style="list-style-type: none"> <li>합계금액에서 사용한 해피포인트를 차감 후 표시 및 결제</li> </ul>	
	10	<b>해피포인트 고객 정보</b> <ul style="list-style-type: none"> <li>소비자의 승인번호와 적립/잔여 포인트 표시</li> </ul>	
	11	<b>바코드</b> <ul style="list-style-type: none"> <li>해당 영수증의 바코드 표시</li> </ul>	

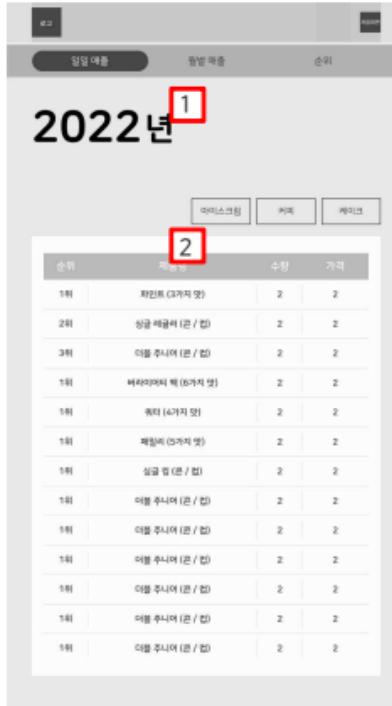
## 2. 기능 명세서

관리자 화면 - 일일 매출	순번	기능명세서	변수
	1	<b>돌아가기</b> <ul style="list-style-type: none"><li>클릭 시 “상품 안내” 화면으로 돌아감</li></ul>	
	2	<b>탭 메뉴</b> <ul style="list-style-type: none"><li>탭을 누르면 각 탭에 맞는 화면이 출력됨</li></ul>	
	3	<b>날짜 표기</b> <ul style="list-style-type: none"><li>당일 날짜를 표기</li></ul>	
	4	<b>매출액 및 판매수량</b> <ul style="list-style-type: none"><li>당일 판매된 제품의 수량 및 매출액 표시</li><li>상품군을 묶어 매출액과 수량을 표시</li><li>세 자릿수마다 콤마 표기</li></ul>	
	5	<b>상품 선택 버튼</b> <ul style="list-style-type: none"><li>각 버튼을 클릭하면 해당 상품군에 따른 통계를 확인할 수 있음</li></ul>	
	6	<b>제품별 수량 통계</b> <ul style="list-style-type: none"><li>클릭한 상품군에 속한 제품들의 이름과 당일 판매한 수량을 출력</li></ul>	

## 2. 기능 명세서

관리자 화면 - 월별 매출	순번	기능명세서	변수
	1	<p><b>월 표기</b></p> <ul style="list-style-type: none"> <li>• 당월 표기</li> <li>• 해당 월을 기준으로 값 표기</li> </ul>	
	2	<p><b>통계 정보</b></p> <ul style="list-style-type: none"> <li>• “일일 매출” 화면과 같이 상품군을 선택하면 해당 상품군의 총 매출액과 판매 수량 표시</li> <li>• 각 상품군에 해당하는 제품의 판매수량 표시</li> </ul>	

## 2. 기능 명세서

관리자 화면 - 순위	순번	기능명세서	변수																																																								
 <p>The screenshot shows a manager ranking screen. At the top, it says '2022년' (Year 2022). Below that is a table with columns: 순위 (Rank), 제품명 (Product Name), 수량 (Quantity), and 가격 (Price). The table lists various products and their sales volume for the year 2022.</p> <table border="1"> <thead> <tr> <th>순위</th> <th>제품명</th> <th>수량</th> <th>가격</th> </tr> </thead> <tbody> <tr><td>1위</td><td>최인트 (3가지 맛)</td><td>2</td><td>2</td></tr> <tr><td>2위</td><td>청글 배글리 (끈 / 뱀)</td><td>2</td><td>2</td></tr> <tr><td>3위</td><td>이불 푸니어 (끈 / 뱀)</td><td>2</td><td>2</td></tr> <tr><td>4위</td><td>버라이어티 박 (5가지 맛)</td><td>2</td><td>2</td></tr> <tr><td>5위</td><td>코피 (4가지 맛)</td><td>2</td><td>2</td></tr> <tr><td>6위</td><td>페플리 (5가지 맛)</td><td>2</td><td>2</td></tr> <tr><td>7위</td><td>실글 뱀 (끈 / 뱀)</td><td>2</td><td>2</td></tr> <tr><td>8위</td><td>이불 푸니어 (끈 / 뱀)</td><td>2</td><td>2</td></tr> <tr><td>9위</td><td>이불 푸니어 (끈 / 뱀)</td><td>2</td><td>2</td></tr> <tr><td>10위</td><td>이불 푸니어 (끈 / 뱀)</td><td>2</td><td>2</td></tr> <tr><td>11위</td><td>이불 푸니어 (끈 / 뱀)</td><td>2</td><td>2</td></tr> <tr><td>12위</td><td>이불 푸니어 (끈 / 뱀)</td><td>2</td><td>2</td></tr> <tr><td>13위</td><td>이불 푸니어 (끈 / 뱀)</td><td>2</td><td>2</td></tr> </tbody> </table>	순위	제품명	수량	가격	1위	최인트 (3가지 맛)	2	2	2위	청글 배글리 (끈 / 뱀)	2	2	3위	이불 푸니어 (끈 / 뱀)	2	2	4위	버라이어티 박 (5가지 맛)	2	2	5위	코피 (4가지 맛)	2	2	6위	페플리 (5가지 맛)	2	2	7위	실글 뱀 (끈 / 뱀)	2	2	8위	이불 푸니어 (끈 / 뱀)	2	2	9위	이불 푸니어 (끈 / 뱀)	2	2	10위	이불 푸니어 (끈 / 뱀)	2	2	11위	이불 푸니어 (끈 / 뱀)	2	2	12위	이불 푸니어 (끈 / 뱀)	2	2	13위	이불 푸니어 (끈 / 뱀)	2	2	1	<b>년도 표기</b> <ul style="list-style-type: none"> <li>해당 년도 표기</li> <li>해당 년도를 기준으로 순위 정렬</li> </ul>	
순위	제품명	수량	가격																																																								
1위	최인트 (3가지 맛)	2	2																																																								
2위	청글 배글리 (끈 / 뱀)	2	2																																																								
3위	이불 푸니어 (끈 / 뱀)	2	2																																																								
4위	버라이어티 박 (5가지 맛)	2	2																																																								
5위	코피 (4가지 맛)	2	2																																																								
6위	페플리 (5가지 맛)	2	2																																																								
7위	실글 뱀 (끈 / 뱀)	2	2																																																								
8위	이불 푸니어 (끈 / 뱀)	2	2																																																								
9위	이불 푸니어 (끈 / 뱀)	2	2																																																								
10위	이불 푸니어 (끈 / 뱀)	2	2																																																								
11위	이불 푸니어 (끈 / 뱀)	2	2																																																								
12위	이불 푸니어 (끈 / 뱀)	2	2																																																								
13위	이불 푸니어 (끈 / 뱀)	2	2																																																								
	2	<b>순위 정렬</b> <ul style="list-style-type: none"> <li>각 상품군의 판매 금액을 기준으로 정렬</li> <li>해당 년도에 어떤 제품이 가장 많이 팔렸는지 알 수 있음</li> </ul>																																																									