

# 마지막 음식을 위한 푸드셰어링 플랫폼



2020.09.14 – 2020.10.08

Team. G ( 강정윤, 김원우, 용수민, 이상우, 한주연 )

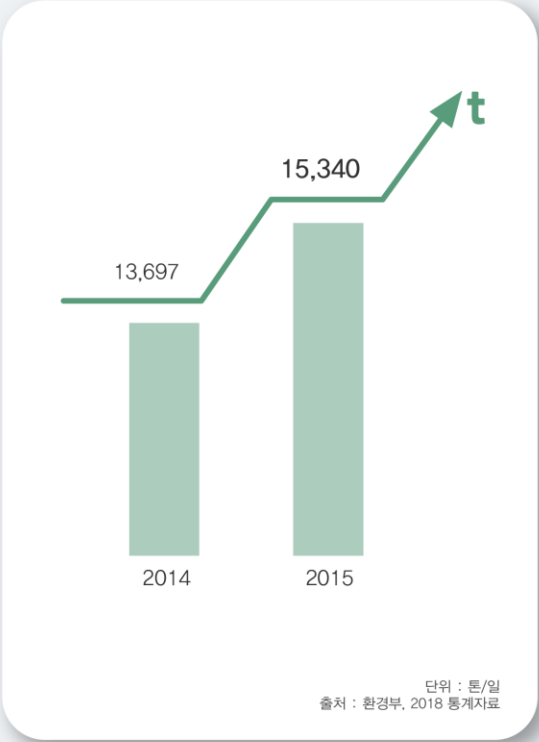


다들 한 번쯤 먹지도 않고

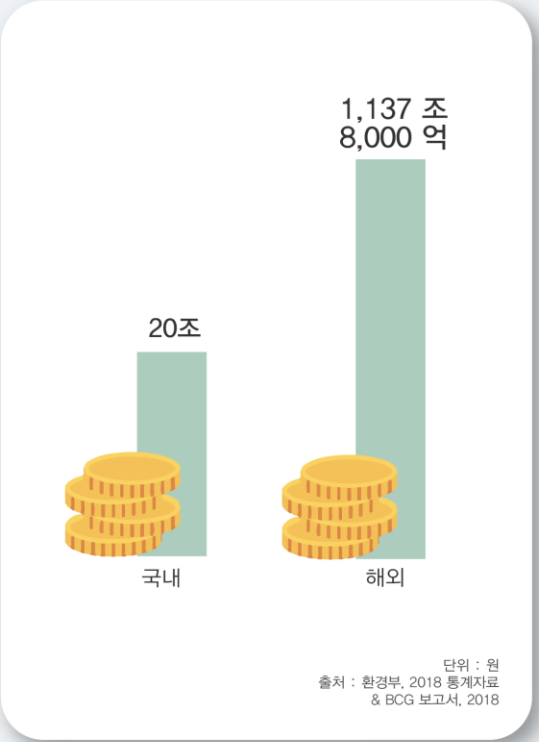
음식물 쓰레기 버려본 적 있으시나요?



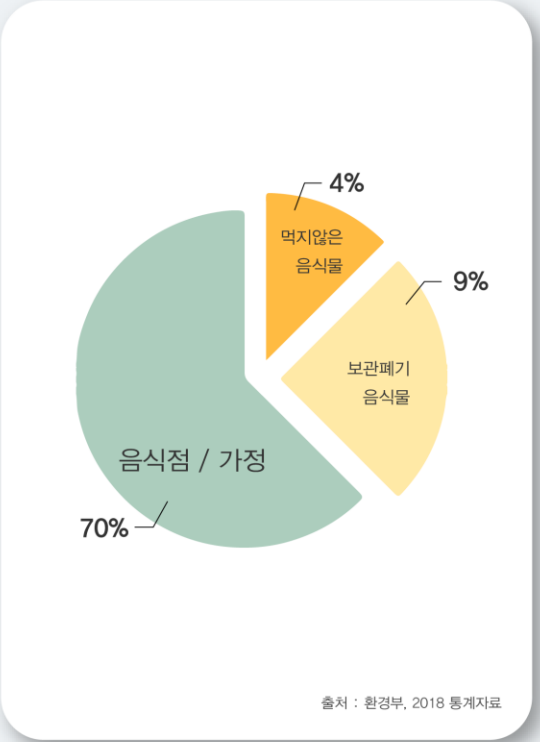
연간 음식폐기물 현황 (국내)



2018년 연간 음식물 처리 비용



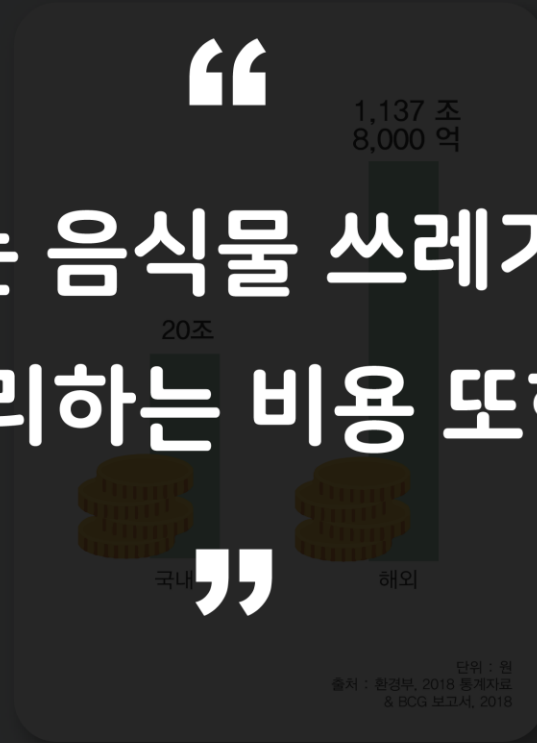
음식물쓰레기 비율



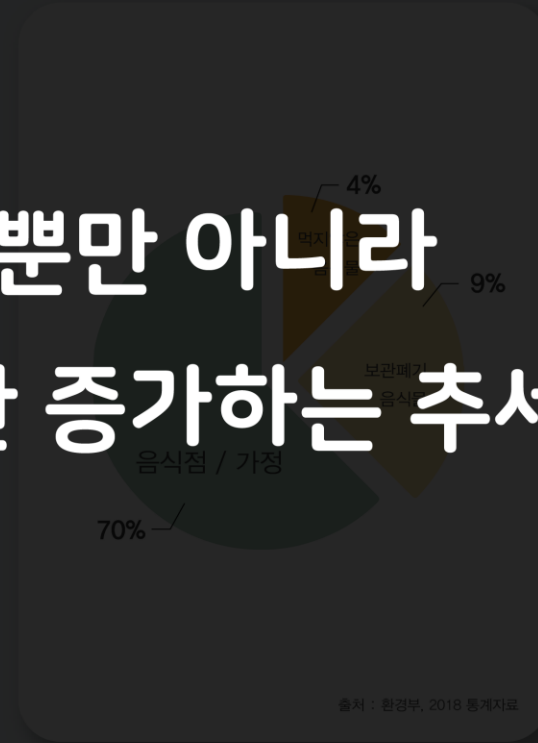
연간 음식폐기물 현황 (국내)



2018년 연간 음식물 처리 비용



음식물쓰레기 비율



“  
먹지 않고 버리는 음식물 쓰레기뿐만 아니라  
음식물 쓰레기를 처리하는 비용 또한 증가하는 추세  
”



## 독일의 공유냉장고 형태의 '푸드 셰어링' 운동



독일의 푸드셰어링 \_ 길거리 냉장고



먹거리 나눔이 끼치는 영향은 아직  
제한적이며, 음식물 쓰레기 방지를 위한  
온전한 해결책이라고는 생각하지 않는다.  
중요한 것은

**“이를 통해 사람들의 생각을  
바꾸는 것”**

푸드셰어링 운동가, 저널리스트  
발렌틴 투른

## 해외와 국내의 다양한 서비스



국내 \_ 라스트오더



덴마크 \_ 투굿투고



스웨덴 \_ 카르마



영국 \_ 올리오

마감할인 서비스

식재료 무료나눔

## 각각의 서비스에 존재하는 문제점들

소비자 간 거래,  
가정 간 식자재 공유 불가

특정 장소와 시각에  
픽업해야하는 불편함

무료나눔으로 인한  
참여동기 저하



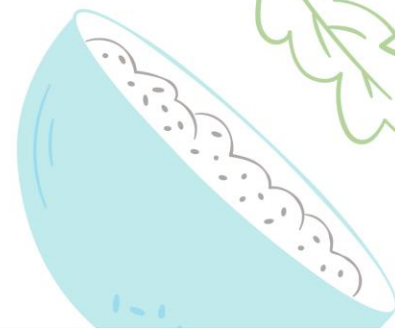
# 마음마켓

블록체인 X

푸드셰어링 커뮤니티 플랫폼



콩한쪽도  
나누어요,  
마음마켓





기존 국내외 서비스들이 가지고 있던  
문제점들을 블록체인으로 해결한 '마음마켓'

## AS-IS

1. 소비자 간의 거래 시 신뢰성 문제

2. 특정 장소와 시간의 제약

3. 무료 나눔으로 인한 참여동기 저하

-----

-----

-----

## TO-BE

사용자들의 참여 활용 인증 및 블록에 정보 기록

채팅 기능을 활용한 쌍방향 소통 기능 확보

인증참여에 대한 포인트 보상 및 인앱 결제시 할인 제공

## 마음마켓의 핵심기능 1

## 마음나누기

아라곤 블록체인 활용  
+ 보상제공 (앱 내 포인트)

<

제품 등록

>

제품 사진을 등록하세요.

사진첨부

제품명

제품명

지역

00구00동

희망가격

0원

해당 제품의 영수증을 등록하세요.

사진첨부

구매일자

YYMMDD

유통기한

YYMMDD

제품 상세설명 (ex. 치고의 당도를 자랑하는 수박 나눔입니다) 최대 1000byt

\* 해당 서비스는 불특정인을 이용하여 투명한 거래를 지원합니다.  
정보 저장 후 수정하실 수 있으며, 허위사실 입력 시 불이익을 받을 수 있습니다.  
정보 확인 후 진행해주세요.

수정하기


등록하기

SELL

블록체인 X 마을마켓

## 사용방법 꿀팁!

더 알아보기

지금 인증에 참여하고 보상받으세요! 

### 호박고구마 2개

인증하기

상인동 320 m 이내

2020.09.27 구매

3/5 참여중

### 상추 반단

인증하기

도원동 1.30 km 이내

2020.09.28 구매

1/5 참여중

### 바나나 반손

인증하기

도원동 1.30 km 이내

2020.09.27 구매

2/5 참여중

### 가래떡 7줄

인증하기

상인동 720 m 이내

2020.09.25 구매

4/5 참여중

### 대파 반단


인증하기

진천동 1.7 km 이내

SELL


<
인증 하기
×



제품명  
**자두 반 박스**

판매지역  
**달서구 유천동**



구매일자  
**20.09.28**

유통기한  
**20.10.15**

\* 인증에 참여하시면 적립금 10원이 제공됩니다.  
인증은 하루 최대 10회 가능합니다.  
적립금은 후주 월급 / 현금기부금 으로 사용됩니다.

### 사진과 위의 내용이 일치합니까?

1  
**비동의**

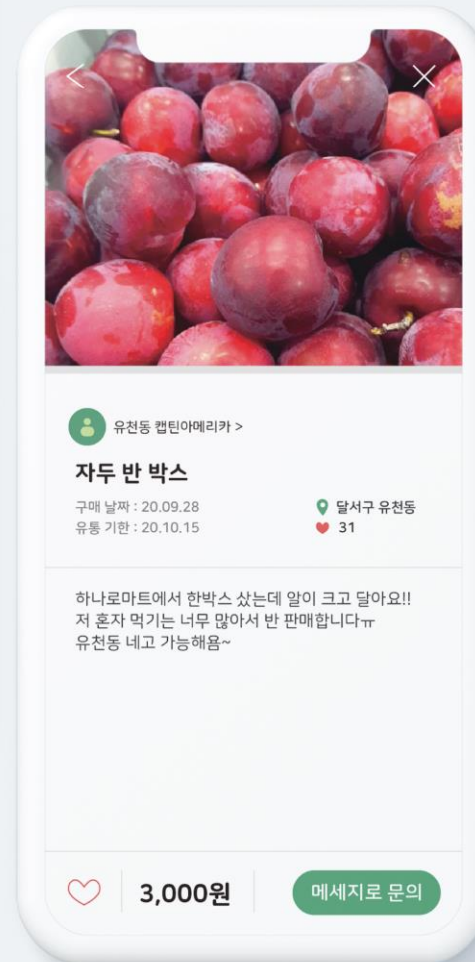
3  
**동의**

## 마음마켓의 핵심기능 2

마음주기

마음받기

C2C 형태의 거래

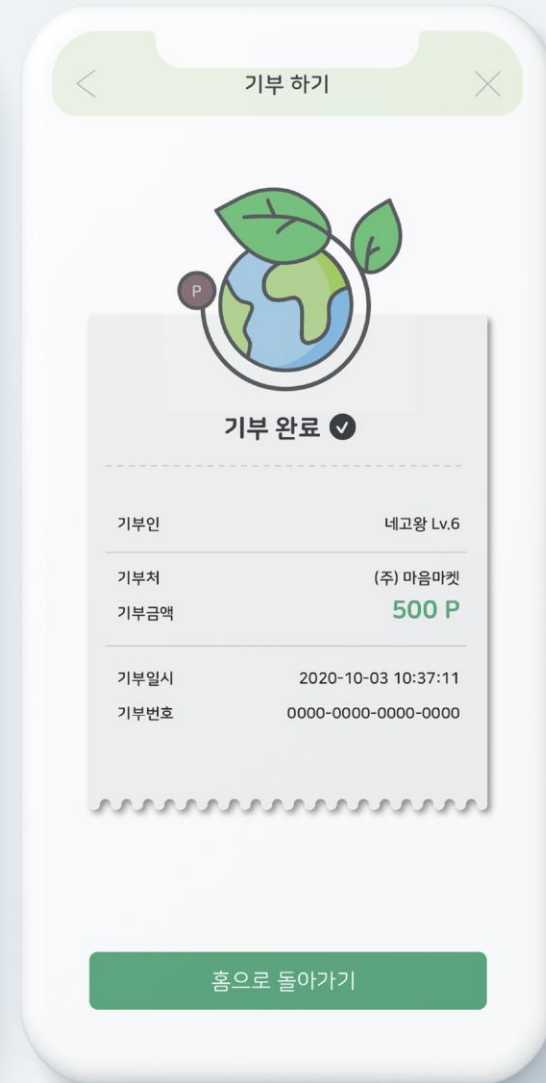




## 마음마켓의 핵심기능 3

### 기부하기

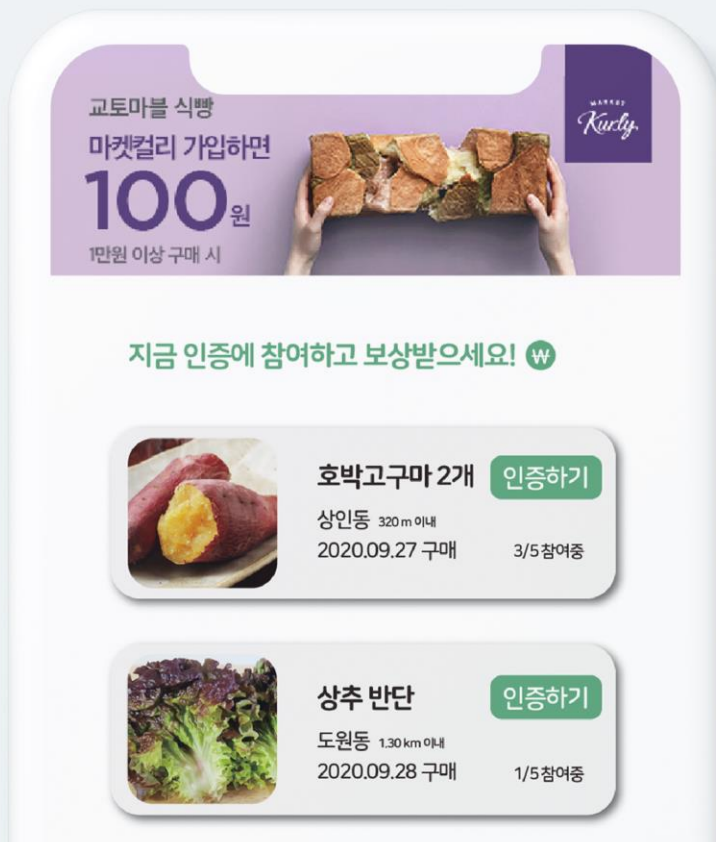
선택에 따라 적립금 기부



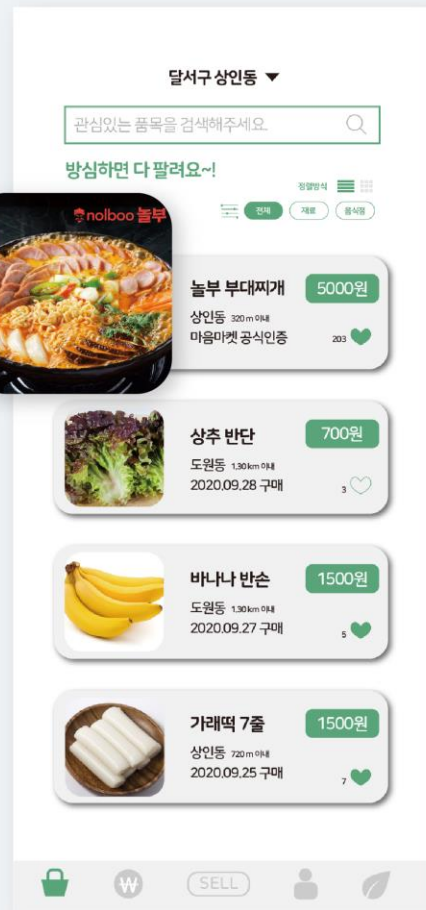
### 3-1 수익모델

#### 투트랙 전략 (멤버십 기능 & 광고)

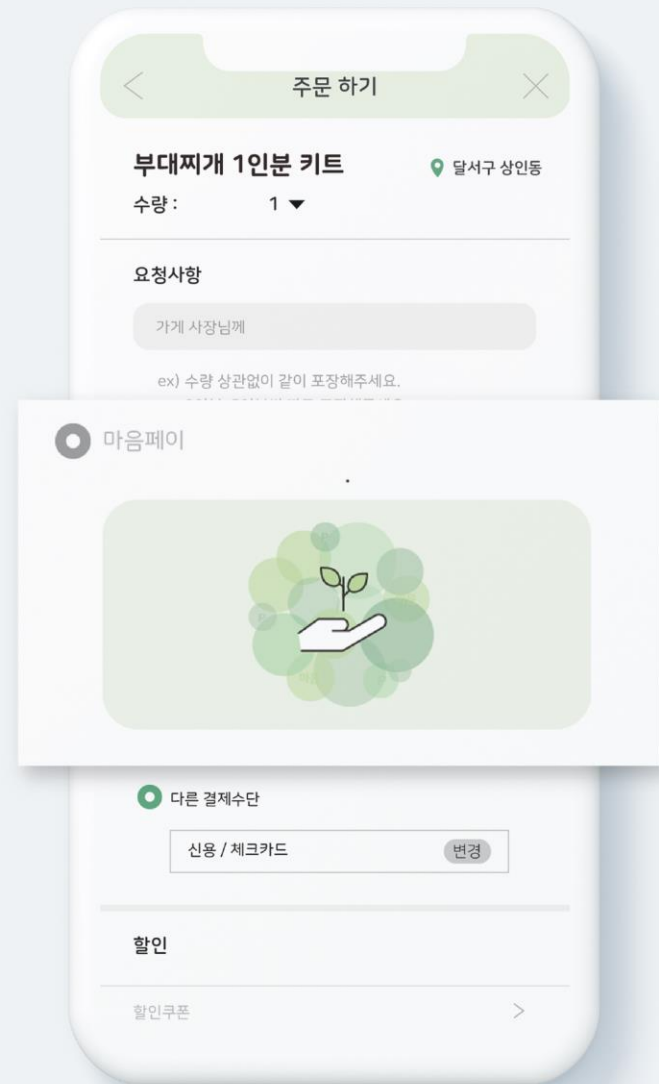
배너광고 예시



인기



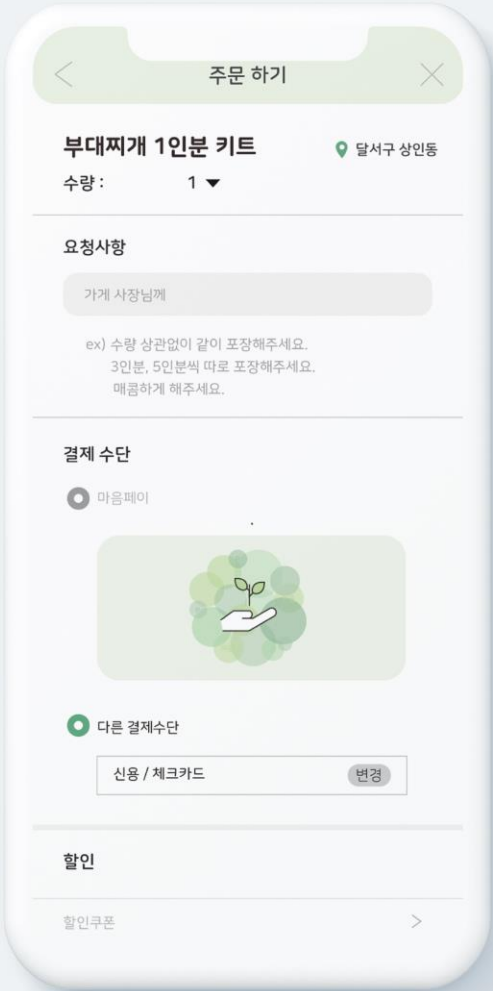
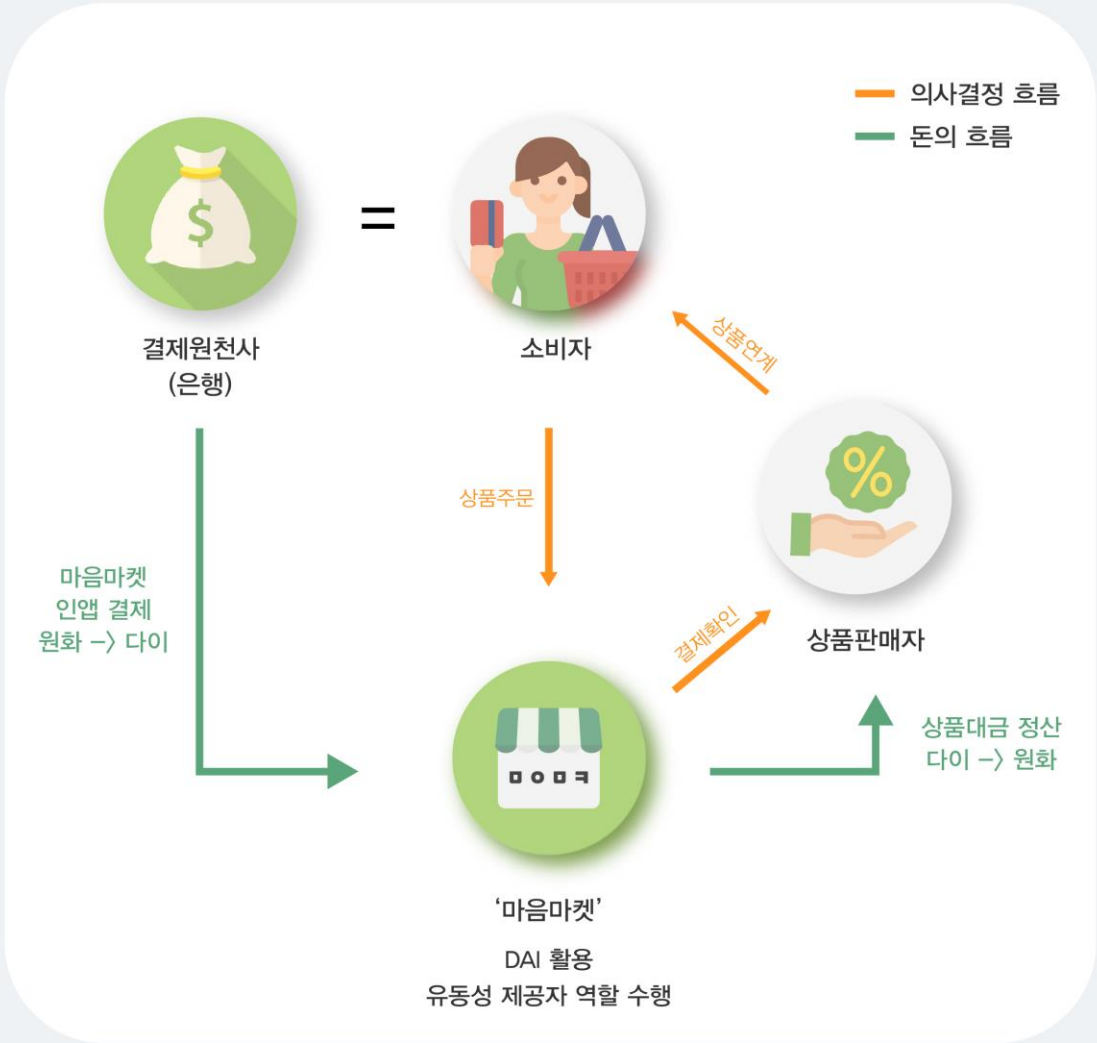
멤버십 혜택  
(상위노출 + 파워링크)



인앱결제

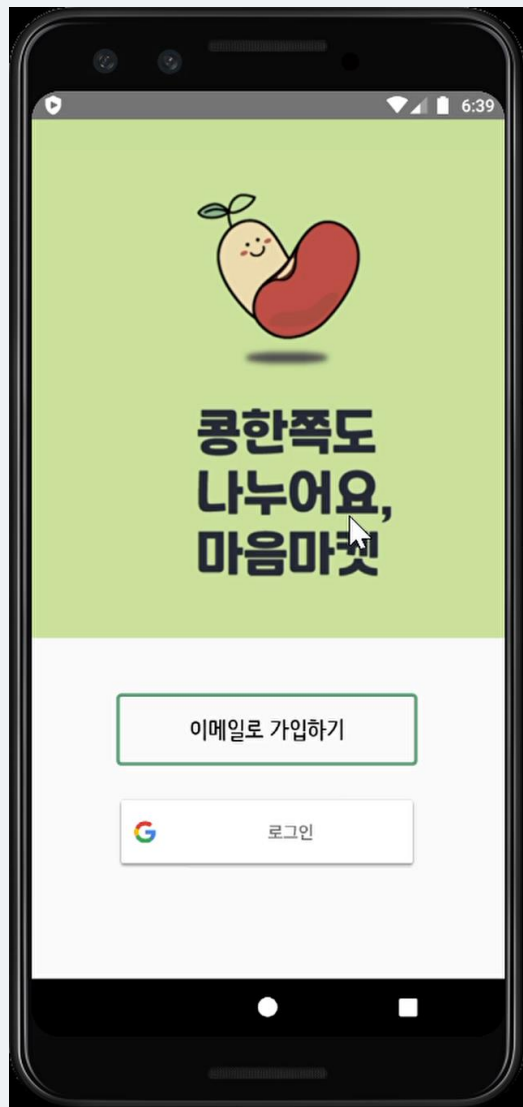
3-1 수익모델

마음마켓이 DAI를 활용하여  
유동성 제동자 역할 수행



인앱결제

### 3-2 데모영상



```
1 pragma solidity >=0.4.22 <0.7.0;
2
3 contract SimpleStorage{
4
5     struct food {
6         uint expiration_year;
7         uint expiration_month;
8         uint expiration_day;
9         uint buy_year;
10        uint buy_month;
11        uint buy_day;
12        string product_name;
13        string person_name;
14        bool verification;
15    }
16
17    food[] private foods;
18
19    mapping (string => address) public foodToOwner;
20
21    function _createfood(uint _expiration_year, uint _expiration_month, uint _expiration_day, uint _buy_year, uint _buy_month, uint _buy_day,
22        string memory _product_name, string memory _person_name, bool _verification) private {
23        require(_verification == true);
24        foods.push(food(_expiration_year, _expiration_month, _expiration_day, _buy_year, _buy_month, _buy_day, _product_name, _person_name, _verification));
25    }
26 }
```

5개 이상의 인증을 받은 음식에 한해 그 정보들을 블록체인에 저장하는 컨트랙트 코드

```
private val FAUCET_ADDRESS = listOf(
    Address("0x8ced5ad0d8da4ec211c17355ed3dbfec4cf0e5b9")
)
val TOKEN_CONTRACT_ADDRESS = Address
Address("0x8ced5ad0d8da4ec211c17355ed3dbfec4cf0e5b9")
val chain = ChainId(5)
val rpc = getMin3RPC(GOERLI_BOOTNODES)
val keyStore = InitializingFileKeyStore(File("keystore")).apply {
    if (getAddresses().isEmpty()) {
        println("creating new keypair")
        addKey(createEthereumKeyPair())
    }
}
```

Etherscan의 컨트랙트 주소와 키페어를 활용하는 코드



3-2 데모영상

Etherscan의 Transaction Details (상세 트랜잭션 정보)

Address 0xA240F065A3954dA438280852a8E5178A1Cb36ddB

Overview

Balance: 4.999418932 Ether

More Info

My Name Tag: Not Available

Transactions

17 Latest 6 from a total of 6 transactions

Txn Hash	Block	Age	From	To	Value	[Txn Fee]
0x0e420197e66d852cd...	8826104	2 hrs 28 mins ago	0xa240f065a3954da43...	OUT Contract Creation	0 Ether	0.000581068
0x5fc38c5a6d3cc58ee...	8826091	2 hrs 32 mins ago	0x81b7e0865bd5d56486...	IN	1 Ether	0.000084
0x4b5253b9b8b0351b...	8826091	2 hrs 32 mins ago	0x81b7e0865bd5d56486...	IN	1 Ether	0.000084
0x2ae45c7fa51142097...	8826091	2 hrs 32 mins ago	0x81b7e0865bd5d56486...	IN	1 Ether	0.000084
0x0405c565c8e3783d...	8826091	2 hrs 32 mins ago	0x81b7e0865bd5d56486...	IN	1 Ether	0.000084
0x1a85799ae42d7ce06...	8826090	2 hrs 32 mins ago	0x81b7e0865bd5d56486...	IN	1 Ether	0.000084



```
buildscript {
    ext.kotlin_version = "1.4.10"
    ext.kethereum_version = "0.83.0"
    repositories {
        google()
        jcenter()
        mavenCentral()
        maven { url 'https://www.jitpack.io' }
    }
    dependencies {
        classpath "com.android.tools.build:gradle:4.0.1"
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"
        classpath 'com.google.gms:google-services:4.3.4'
    }
}
allprojects {
    repositories {
        google()
        jcenter()
        maven { url "https://maven.google.com" }
        maven { url 'https://javadoc.jitpack.io' }
    }
}
task clean(type: Delete) {
    delete rootProject.buildDir
}
```

```
apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'
apply plugin: 'kotlin-android-extensions'
apply plugin: 'com.google.gms.google-services'
android {
    compileSdkVersion 30
    buildToolsVersion "30.0.2"
    defaultConfig {
        applicationId "com.example.foj_project"
        minSdkVersion 26
        targetSdkVersion 30
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}
dependencies {
    implementation fileTree(dir: "libs", include: ["*.jar"])
    implementation "org.jetbrains.kotlin:kotlin-stdlib:$kotlin_version"
    implementation 'androidx.core:core-ktx:1.3.2'
    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.2'
    implementation 'com.google.android.material:material:1.2.1'
    implementation 'androidx.lifecycle:lifecycle-extensions:2.2.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
    implementation 'com.google.firebase:firebase-analytics:17.5.0'
    implementation 'com.google.firebase:firebase-auth:19.4.0'
    implementation 'com.google.android.gms:play-services-auth:18.1.0'
    implementation 'com.github.walleth.kethereum:etherscan:0.83.0'
}
```

KETHEREUM과의 연동을 위해 넣은 코드들



### 타겟시장

마감할인 및 라스트오더 실제 키워드 검색량이 높은 2535 남녀 타겟

1순위 : 강남구, 마포구, 송파구

2순위 : 영등포구, 종로구, 중구

3순위 : 서초구, 강서구, 관악구, 동대문구

통계청 소상공인 및 1인가구 기준

### 유효시장

수도권 1인가구 및 마감 시 재고가 남는 소상공인

### 전체시장

식재료 및 음식 셰어링(마감음식 재고 처리) 희망하는 모든 사람 및 가게

지역기반 커뮤니티 서비스



지역기반 게더링 및 정보공유  
커뮤니티 플랫폼으로  
서비스 확대

비대면 채널 확보



편의점 및 대형마트 협업 유희공간 활용

평판시스템 도입



평판시스템 도입,  
NFT활용 보상 제공

