



포팅매뉴얼

사용 툴 및 버전

- **EC2**
 - Ubuntu 20.04.6
- **Docker**
 - version 24.0.6,
- **웹 서버**
 - nginx/1.18.0 (Ubuntu)
- **Jenkins**
 - Docker 기반
 - 이미지 : jenkins/jenkins:jdk17 기반 사용
 - 구동하는 springboot의 jdk가 17이기 때문에 jenkins도 맞춰주었음
- **JDK**
 - openjdk version "17.0.8" 2023-07-18
- **Gitlab**
 - 16.0.5
- **Visual Studio Code 1.82** - 프론트(Next.js, React, Typescript) 개발
 - nodeJs : 18.17.0
- **IntelliJ IDEA 2023.1.3** - 백엔드 개발
 - openJDK version "17.0.8"
 - gradle 8.2.1

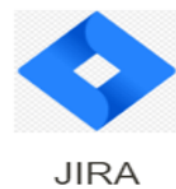
사용 기술 스택

인프라



- EC2 (Ubuntu 20.04.6)
- docker (24.0.5)
- jenkins (jenkins/jenkins:jdk17)
- nginx (1.18.0 Ubuntu)
- S3

관리 툴



백엔드



- Springboot 3.1.3
- SpringSecurity 6.1.3
- MariaDB 11.1.2
- Redis 7.2.1
- QueryDSL 5.0.0

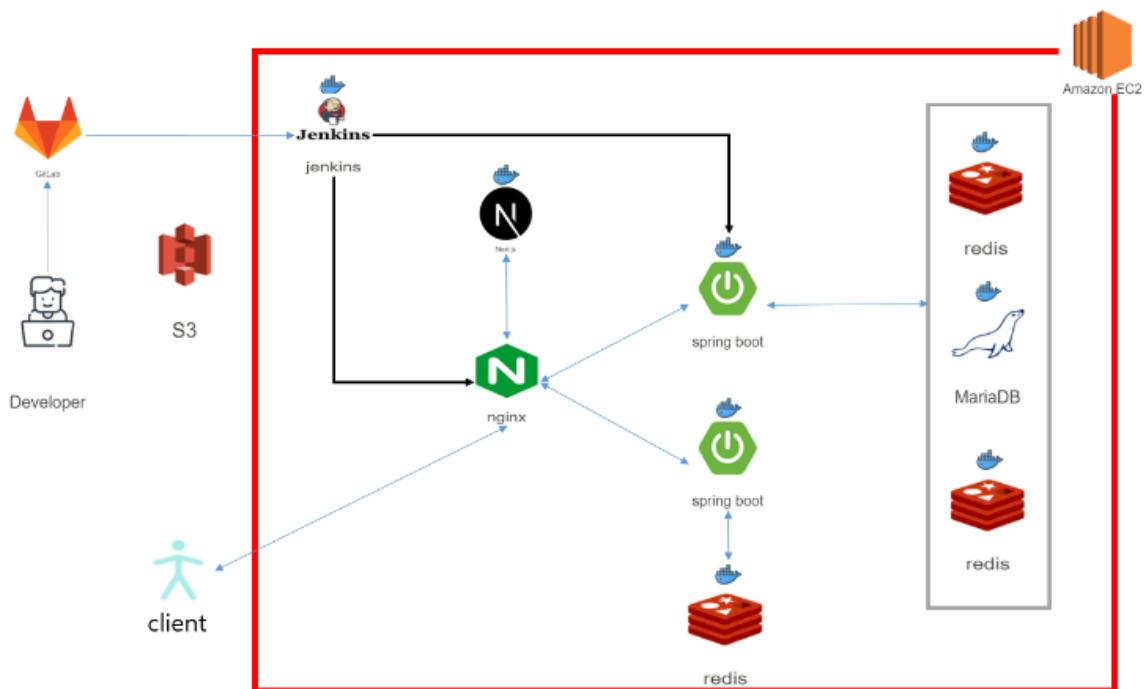
프론트엔드



- Next.js 13. 4.19
- Typescript 5.1
- Tailwind CSS 3.3.2
- React Query 4.33
- Jotai 2.4.1
- Framer-Motion 10.16.4





인프라 구조

개발 서버와 운영 서버의 구조는 동일



포트 정보 및 properties는 보안 상 생략합니다.

- nginx를 제외하고 나머지 docker container 구성
- Jenkins는 설정한 파이프라인에 따라 push 시 백엔드, 프론트 빌드 및 배포가 일어난다.

		moic-backend	9 hr 27 min	#124
		moic-frontend	26 min	#129

- Next.js와 연동하기 위해 별도의 **Node Server** 실행
- Springboot 서버는 2개를 운용
 - 하나는 Main 서버,
 - 하나는 검색어 자동완성 기능을 담당하는 서버
 - 우리의 모든 서비스는 **검색을 통해** 이루어진다.
 - **검색어 자동완성 간 api 호출이 많이 일어나기 때문에** 분리하였다.
- **nginx에서 api의 경로에 따라 로드밸런싱한다.**
 - ssl 적용은 python3의 certbot 이용했음. 제대로 적용이 된다면
 - #managed by Certbot으로 보이는 문장들이 '자동으로' 추가된다.

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;
    if ($host = www.moic.site) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    if ($host = moic.site) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    server_name moic.site www.moic.site;
    return 404; # managed by Certbot
}

server {
    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/moic.site/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/moic.site/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
    server_name moic.site www.moic.site;
    location / {
        proxy_pass http://127.0.0.1:{프론트서버포트};
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        try_files $uri $uri/ / =404;
    }
}
```

```

location /api/v1/autocomplete {
    proxy_pass http://localhost:{백엔드서버1포트};
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
}

location /api/v1 {
    proxy_pass http://localhost:{백엔드서버2포트};
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
}
}

```



이런 개념으로 Load Balancing을 수행하도록, 도메인 별 MSA로 만들 수 있다

- MariaDB : 원본 데이터 저장소
- Redis는 3개
 - mainRedis : 카드, 기프티콘 등 서비스에 필요한 정보의 캐싱을 담당하는 서버
 - SecurityRedis : 리프레시 토큰 저장소
 - autoRedis : 자동완성 단어 셋 저장소
 - 파일 크기가 크지 않고, api 호출 시 DB를 거치는 것보다 메모리에 적용해 두는게 옳다고 생각했다.
 - 둘의 시간 차이는 **10배 가량 난다.**
 - **Z-Set 자료구조**를 이용하여 구현

외부 서비스 정리

OAuth

소셜 로그인

사용자의 회원가입 및 로그인 시 편의성을 위하여 **Google**, **Kakao** 를 이용한 소셜 로그인 사용

build.gradle

```
implementation 'org.springframework.boot:spring-boot-starter-oauth2-client'
```

application.properties

```
#Google
spring.security.oauth2.client.registration.google.client-name=google
spring.security.oauth2.client.registration.google.client-id={Google_ID}
spring.security.oauth2.client.registration.google.client-secret={Google_SECRET}
spring.security.oauth2.client.registration.google.scope=profile,email
spring.security.oauth2.client.registration.google.redirect-uri={Google_REDIRECT_URI}

#Kakao
spring.security.oauth2.client.registration.kakao.client-name=kakao
spring.security.oauth2.client.registration.kakao.client-id={Kakao_ID}
spring.security.oauth2.client.registration.kakao.client-secret={Kakao_SECRET}
spring.security.oauth2.client.registration.kakao.scope=profile_nickname, account_email
spring.security.oauth2.client.registration.kakao.redirect-uri={Kakao_REDIRECT_URI}
spring.security.oauth2.client.registration.kakao.authorization-grant-type=authorization_code
spring.security.oauth2.client.registration.kakao.client-authentication-method=client_secret_post
spring.security.oauth2.client.provider.kakao.authorization-uri=https://kauth.kakao.com/oauth/authorize
spring.security.oauth2.client.provider.kakao.token-uri=https://kauth.kakao.com/oauth/token
spring.security.oauth2.client.provider.kakao.user-info-uri=https://kapi.kakao.com/v2/user/me
spring.security.oauth2.client.provider.kakao.user-name-attribute=id
```

구글 개발자 센터

카카오 개발자 센터

1. 각 개발자 센터에서 애플리케이션 생성 후 발급받은 ID와 SECRET을 properties에 등록합니다.
2. 인증을 받은 redirectURI와 제공받을 사용자 정보의 범위 또한 등록합니다.
3. Google의 경우 Spring Security에서 OAuth 과정이 원활히 진행되도록 세팅 되어 있지만, Kakao의 경우 추가적인 설정을 properties에 작성해 주어야 합니다.

KAKAO MAP

1. 카카오 개발자 센터에서 애플리케이션 생성 후 발급받은 API KEY를 발급받습니다.
2. API KEY로 필요한 API를 호출해 사용합니다.
 - a. KAKAO MAP API
 - b. KAKAO 다중 경유지 길찾기 API


NAVER OCR



기프트콘 이미지에서 텍스트를 추출하기 위해 OCR 서비스 이용

CLOVA OCR Custom API

콘텐츠 공유

 <https://api.ncloud-docs.com/docs/ai-application-service-ocr-ocr>

- 버전에 맞게 공식문서를 참고하고 따라하면 된다. 정말로 그냥 따라하면 된다..
- 홈페이지에서 APIURL과 SECRET을 받고 연동한다.

CHATGPT


```
#ChatGpt
CHAT_APIKEY=
CHAT_APIURL=
CHAT_MODEL=gpt-3.5-turbo
```



공식문서를 반드시 참고할 것!

OpenAI Platform

Explore developer resources, tutorials, API docs, and dynamic examples to get the most out of OpenAI's platform.

 <https://platform.openai.com/docs/introduction>

OpenAI Platform



- ChatGPT와 연동하기 위해서 필요한 정보
 - openAI에서 3개월이 지난 계정의 경우 결제해야 이용할 수 있음
- apiKey 및 apiURL 발급
- model은 본인이 선택한 모델로 설정해줘야 한다.
- ChatGPT의 API를 호출할 때, **형식에 맞게 보내줘야 하기 때문에 커스터마이징이 필요**
- 보내야 하는 형식 → **역시 공식문서에 잘 나와있다.**

```

1  {
2      ··"model": "gpt-4",
3      ··"max_tokens": 100,
4      ··"stream": false,
5      ··"messages": [
6          {
7              ··"role": "user",
8              ··"content": "잘 송이 됐나요?"
9          }
10     ]
11 }

```

```

public class ChatGptResuest {

    private String model;

    private boolean stream;

    private int max_tokens;

    List<ChatGptMessage> messages;

    no usages 김성재
    public ChatGptResuest(String model, boolean stream, int max_tokens, List<ChatGptMessage> messages) {
        this.model = model;
        this.stream = stream;
        this.max_tokens = max_tokens;
        this.messages = messages;
    }
}

```

```

@Getter
@Setter
@Builder
public class ChatGptMessage {

    String role;
    String content;
}

```

- 위와 같이 클래스를 2개 커스텀하여 보낼 수 있게 한다.
- 요청에 성공하면, 응답은 아래와 같이 오게 된다.


```

"model": "gpt-4-0613",
"choices": [
  {
    "index": 0,
    "message": {
      "role": "assistant",
      "content": "확인하는 방법이 따로 제공되지 않아 확인이 불가능합니다. 전송하셨다면 잘 전송된 것으로 보입니다. 만약 전송이 제대로 이루어지지 않았다면, 원하시는 시스템 또는 서비스의 고객 지원센터에 문의해 보시는 것"
    },
    "finish_reason": "length"
  }
]

```

- 우리 서비스에서는, **content** 만 가져오면 된다.
 - 그래서, 저 데이터에서 choices → message → content를 추출하는 메서드를 서버에서 작성하면 된다

S3

- build.gradle 추가

```
implementation 'org.springframework.cloud:spring-cloud-starter-aws:2.2.6.RELEASE'
```

- S3Config
 - AWS S3와 연결하는 작업.

```

@Configuration
public class S3Config {

    @Value("${cloud.aws.credentials.access-key}")
    private String accessKey;
    @Value("${cloud.aws.credentials.secret-key}")
    private String secretKey;
    @Value("${cloud.aws.region.static}")
    private String region;

    @Bean
    public AmazonS3Client amazonS3Client() {
        BasicAWSCredentials awsCredentials = new BasicAWSCredentials(accessKey, secretKey);
        return (AmazonS3Client) AmazonS3ClientBuilder.standard()
            .withRegion(region)
            .withCredentials(new AWSSStaticCredentialsProvider(awsCredentials))
            .build();
    }
}

```

- access-key 및 secret-key를 발급받은 값을 넣어주고
- region도 설정한 값으로 넣어줘서 등록하는 작업이다.

- 이후 필요한 메서드는 커스텀해서 작성한다.
- 업로드 메서드 코드 예시

```
try {  
    amazonS3Client.putObject(new PutObjectRequest(bucket, uniqueName, multipartFile.getInputStream()  
        , objectMetadata).withCannedAcl(CannedAccessControlList.PublicRead));  
} catch (IOException e) {  
    throw new NotFoundException(ExceptionEnum.GIFTCARD_NOT_FOUND);  
}
```

- putObject 메서드를 통해 통해 넣어줄 정보를 입력
 - 나의 bucket이름을 넣어주고, 해당 파일에 대해 설정할 이름(uniqueName)을 설정해서 준다.

DB 덤프 파일 **exec 폴더 안에 업로드**

시연 시나리오



로그인

더 많은 서비스를 누려보세요

로그인

아이디 혹은 비밀번호를 잊으셨나요?

또는

카카오톡로 로그인하기

구글로 로그인하기

혹시 회원가입을 하지 않으셨나요?

1. 최초에는 로그인 페이지로 이동하게 됩니다.

로그인은 **회원가입/카카오/구글** 세가지 중 하나를 택할 수 있습니다.

시연에서는 회원가입을 통한 로그인을 해보겠습니다.

회원가입

약관 동의하기

개인정보 처리방침

핀프들(이하 "회사"라 함)은 개인정보보호법을 준수하며, 관련 법령에 의거한 개인정보처리방침을 정하여 이용자 권익 보호에 최선을 다하고 있습니다. 회사의 개인정보처리방침은 다음과 같은 내용을 담고 있습니다.

1. 개인정보의 처리 목적 및 수집 항목

이전으로

다음으로

회원가입

개인정보 입력

이름

이메일

성별

년도

올바른 이메일을 입력해주세요. ✓

선택안함 <

선택안함 <

이전으로

다음으로

회원가입

계정 정보 입력

아이디

비밀번호

비밀번호 확인

6자 이상 ~ 12자 이하, 영문(소문자), 숫자, 특수문자 조합을 입력해주세요. ✓

비밀번호는 8~16자의 영문(대/소문자), 숫자, 특수문자 조합입니다. ✓

비밀번호와 같은 값을 입력해주세요.

이전으로

회원가입 하기

2. 회원가입 페이지는 위와 같습니다.

성별과 년도를 제외하고는 모두 필수로 응답해야 합니다.

아이디 / 비밀번호 찾기

인증을 통해 계정 정보를 찾아보세요.

아이디 찾기

비밀번호 찾기

아이디 찾기

인증을 통해 계정 정보를 찾아보세요.

이름

이메일

올바른 이메일을 입력해주세요. ✓

아이디 찾기

비밀번호 찾기

인증을 통해 계정 정보를 찾아보세요.

이름

아이디

이메일

6자 이상 ~ 12자 이하, 영문(소문자), 숫자, 특수문자 조합을 입력해주세요. ✓

올바른 이메일을 입력해주세요. ✓

인증요청

3. 만약 아이디와 비밀번호를 잊어버렸다면, 찾기가 가능합니다.

비밀번호 찾기의 경우에는 이메일 인증 번호 발송을 통한 본인 인증 후 비밀번호 변경이 가능합니다.

모익

test



[절약 Tip] 너무 자주 씻지 말아요!



지도 확인하기

지도에서 혜택 지역을 찾아
볼까요?



경로 추천받기

오늘은 어디갈까? 저희가
추천해드릴게요!



기프티콘

기프티콘을 등록하고 관리
해보세요



혜택 확인하기

사용하는 카드를 등록하고
혜택을 확인해보세요!



모익

test



계정 관리

북마크 관리

문의사항

로그아웃

[절약 Tip] 너무 자주 씻지 말아요!



지도 확인하기

지도에서 혜택 지역을 찾아
볼까요?



경로 추천받기

오늘은 어디갈까? 저희가
추천해드릴게요!



기프티콘

기프티콘을 등록하고 관리
해보세요



혜택 확인하기

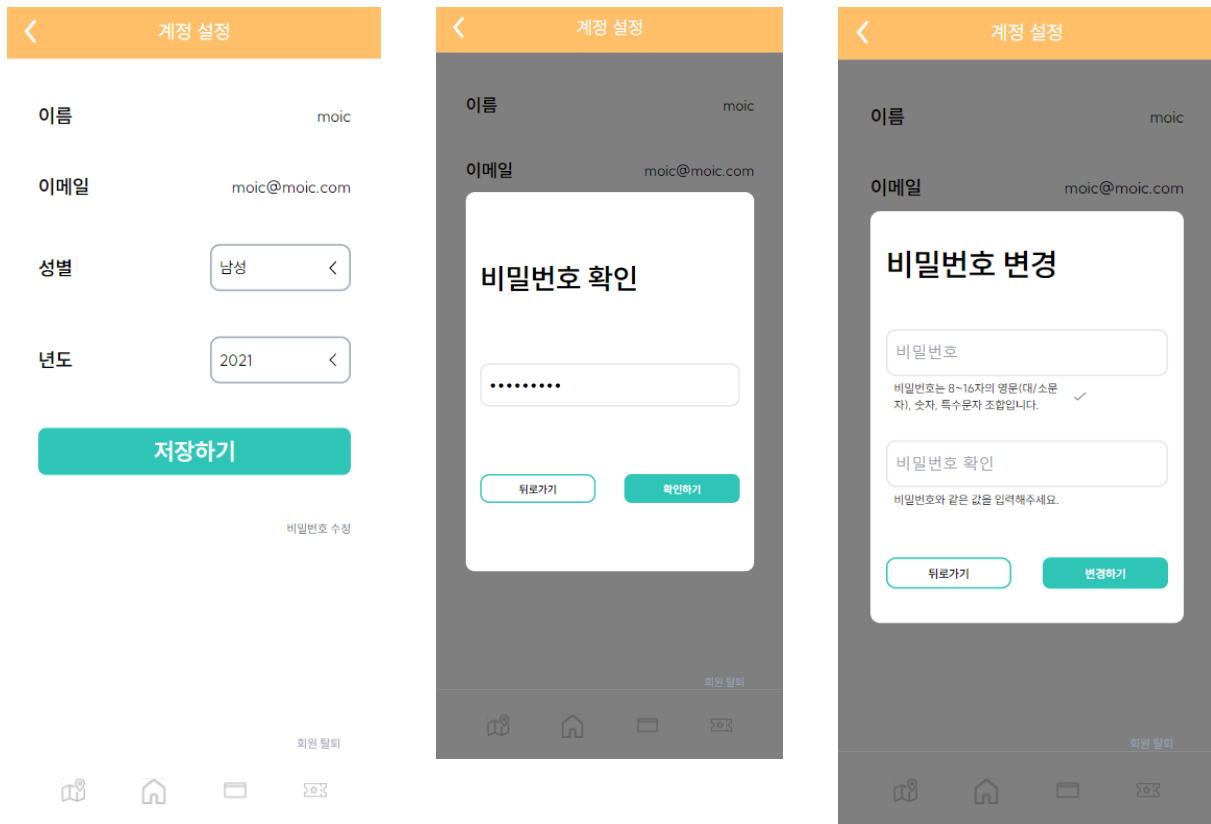
사용하는 카드를 등록하고
혜택을 확인해보세요!



4. 로그인 이후 페이지입니다.

홈에서는 [지도/경로 추천](#)/[기프티콘/카드](#) 페이지로 이동할 수 있습니다.

드롭바에서는 [마이페이지/북마크 관리/VOC](#) 로 이동할 수 있습니다.



5. 마이페이지에서는 [회원정보 수정](#)/[비밀번호 수정](#)/[회원 탈퇴](#) 가 가능합니다.

수정하기

스타벅스 낙성대역점 커피전문점

서울특별시 관악구 남부순환로 1908

문의사항 건의

모역을 사용하셨을 때, 느꼈던 불편한 점,
개선할 점, 관리자에게 문의할 점을 편하
게 적어주세요.(최대 500자)

뒤로가기

건의하기

< 1 >

📍

🏠

📁

🔍

📍

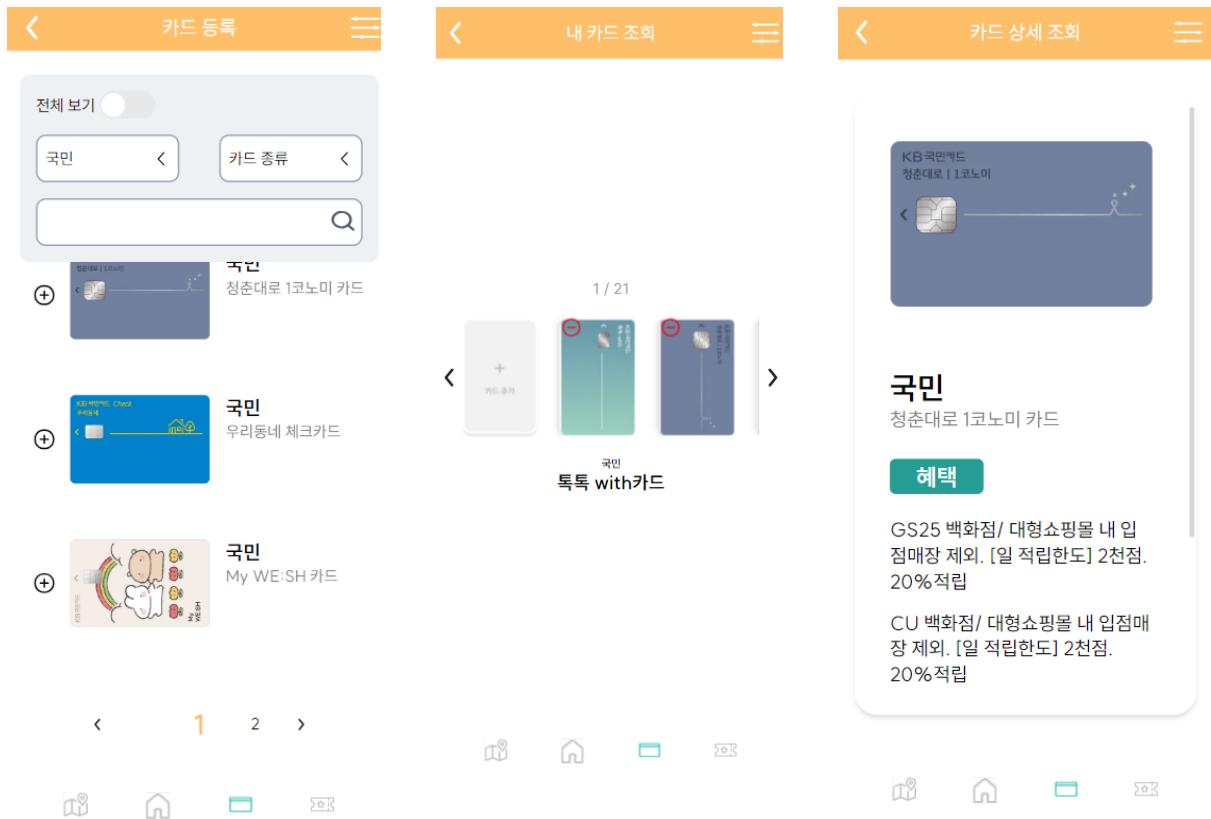
🏠

📁

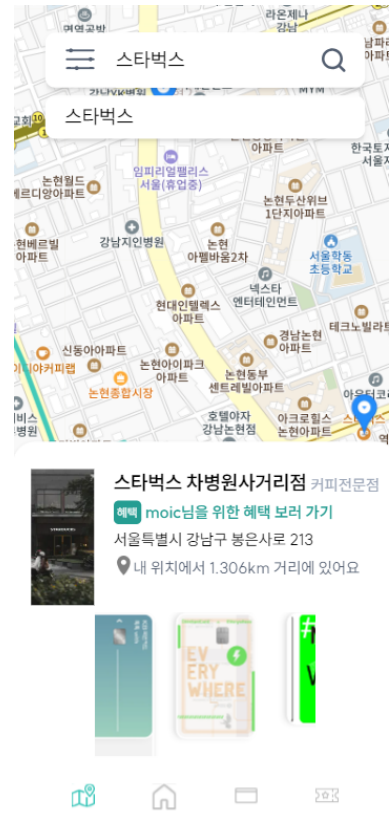
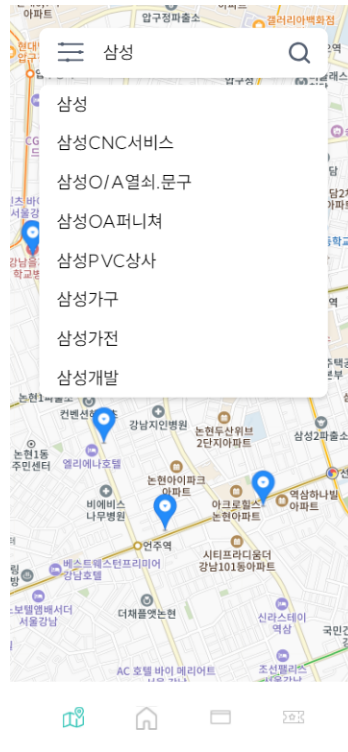
🔍

6. **북마크 조회**의 경우, 내가 설정한 북마크 리스트를 확인하고 수정할 수 있습니다.

VOC 등록의 경우, 건의한 내용이 관리자 이메일로 발송됩니다.



7. **카드** 페이지에서는 전체 카드 조회/내 카드 추가 및 삭제/카드별 혜택 확인 이 가능합니다.



8. 지도 페이지에서는 카테고리별 조회/혜택 가맹점 검색/가맹점 상세조회가 가능합니다.
혜택 보러 가기를 클릭하면 카드와 혜택을 확인할 수 있습니다.



9. 기프트콘 페이지에서는 내 기프트콘 조회/등록/삭제/상세조회가 가능합니다.
기프트콘 등록의 경우, 이미지 업로드를 통해 이루어집니다.