

# Aphantasia 증후군 환자를 위한 Generative AI를 활용한 모바일 어플리케이션 구현

Implementation of Mobile Application  
using Generative AI for Aphantasia Syndrome Patients

2019102231 조수연

## 요약

아판타시아(Aphantasia) 증후군이란 이미지를 머릿속으로 상상하거나 구현할 수 없는 인지장애이다. 이러한 아판타시아 증후군을 가진 사람들이 독서를 할 때 특정 객체를 시각화하지 못하는 문제를 해결하기 위해, Diffusion Model 을 활용한 모바일 어플리케이션을 구현한다.

## 1. 서론

### 1.1 연구 배경

아판타시아(Aphantasia) 증후군이란 이미지를 머릿속으로 상상하거나 구현할 수 없는 인지장애이다. 사고력은 정상이나 심상만이 보이지 않는 것이 특징이다. 전 세계 인구의 2.5% 정도가 이 증후군을 가지고 있다고 추정되며, 스스로 증후군을 지니고 있음을 인지하기 어렵다. 또한, 시각적 정보를 기억하거나 새로운 개념을 상상하는 능력에 영향을 줄 수 있다. 이 상태는 사람의 일상 생활에 영향을 미칠 수 있으며 탐색, 인식 또는 통신과 같은 특정 작업에 어려움을 초래할 수 있다. 따라서 독서와 같이 상상을 바탕으로 시각화를 하는 작업에 어려움을 겪는다.

본 연구에서는 이러한 아판타시아 증후군을 가진 사람들이 독서를 할 때 특정 객체를 시각화하지 못해 독서에 어려움을 겪는 문제를 해결하기 위해, text-to-image 생성 모델을 활용한 모바일 어플리케이션을 제작한다. 이를 사용하여 독서를 하는 과정에서 발견한 객체에 대한 시각 정보를 제공하여 아판타시아 증후군 환자들의 독서를 돋는다.

### 1.2 연구 목표

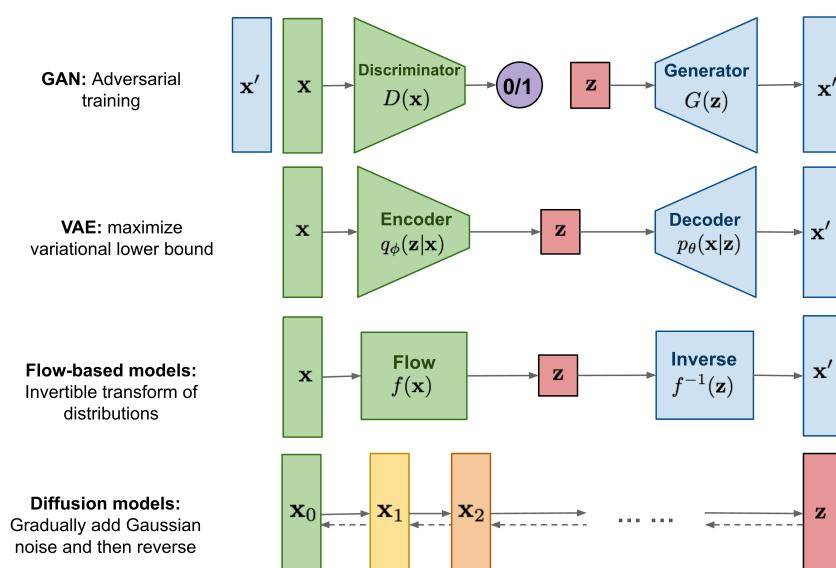
본 연구는 아판타시아 증후군 환자를 위한 모바일 애플리케이션 개발에서 생성 모델의 사용을 탐구한다. 생성 모델은 아판타시아 증후군을 가진 사람들이 세상과 상호 작용하는 새로운 방법을 제공할 수 있는 잠재력을 가지고 있다. 생성 모델을 사용하여 text-to-image를 통해 책의 글에서부터 이미지를 생성하여 환자가 새로운 방식으로 세상을 경험할 수 있도록 돋는다. 이는 기존 데이터에서 시각적 패턴이나 소리를 학습하고 복제할 수 있는 VAE(변형 자동 인코더) 또는 GAN(생성적 적대 신경망)과 같은 딥러닝 기술을 사용하여 달성을 할 수 있다.

제안된 모바일 애플리케이션은 아판타시아 증후군 환자가 독서를 할 때 새로운 개념을 상상하고 특정 물체를 시각화할 수 있는 능력을 향상시키기 위해 ARKit을 사용한 3D 객체 시각화 플랫폼을 제공하는 것을 목표로 한다. 또한 책의 글을 쉽게 가져오기 위해 OCR 기능을 함께 구현하여 편리하게 사용할 수 있도록 사용성을 높인다.

## 2. 관련 연구

### 2.1 생성 모델 (Generative Model)

생성모델은(Generative Model) 데이터를 입력 받아, 입력 받은 데이터와 유사한 분포를 따르는 새로운 데이터를 생성하는 모델이다. 학습은 주어진 데이터의 분포를 얼마나 잘 학습하는지를 고려하며 이루어진다. 생성 모델의 종류로는 Variational AutoEncoder, Generative Adversarial Network(GAN), Flow Model, Diffusion Model 등 다양한 모델이 존재한다.



[그림 1] 다양한 생성 모델의 개요

생성 모델은 크게 지도적, 비지도적 두 가지 종류로 나뉜다. 지도적 생성모델은 레이블이 있는 데이터에 대해서 각 클래스 별 특징 데이터의 확률분포  $P(X|Y)$ 를 추정한 다음 베이즈 정리를 사용하여  $P(Y|X)$ 를 계산한다.  $P(Y|X)$ 를 계산할 수 있기 때문에 분류 모델로 활용할 수 있으며 클래스별 Conditional 확률  $P(X|Y)$ 를 추정했기 때문에 확률분포 상에서의 새로운 가상의 데이터를 생성하거나 확률분포 끝자락에 있는 데이터를 이상치로도 판단하는 이상치 판별 모델로도 활용할 수 있다. 비지도적 생성모델은 다시 통계적 모델과 딥러닝을 활용한 모델로 나뉜다. 비지도적 생성 모델은 레이블이 없다. 따라서 데이터 X 자체의 분포를 학습하여 X의 모분포를 추정하는 학습데이터의 분포를 학습하는 모델이다. GAN기반의 모델은 이에 해당한다.

## 2.2 Text-to-Image 생성 모델

Text-to-Image 생성 모델이란 텍스트가 입력으로 주어졌을 때 해당 텍스트 설명에 부합하는 이미지를 생성하는 인공지능 모델이다. 높은 품질의 이미지를 생성하는 동시에 입력으로 주어진 텍스트를 알맞게 반영하는 것이 중요하다. 2021년 1월 OpenAI 가 발표한 DALL-E의 등장부터 지금까지 가장 빠르게 발전하고 있는 분야 중 하나이다. 기존 text-to-image 생성 모델과 최근 각광받는 모델의 주요한 차이 중 하나는 바로 학습 데이터의 크기에서 나타난다. 최신 모델들은 text와 image 쌍으로 이루어진 초거대 데이터셋으로 학습되었기에 기존 생성 모델보다 다양한 장면을 생성할 수 있고 텍스트가 담고있는 의미도 알맞게 반영한다.

대표적인 Text-to-Image 생성 모델인 DALL-E 의 동작 원리는 다음과 같다. 먼저 텍스트와 시각적 Semantics를 연결한다. CLIP 모델 이용하여 수많은 이미지와 관련 캡션을 이용하여 해당 이미지와 캡션의 연관성을 학습한다. 시각적 Semantics로부터 이미지 생성한 후, GLIDE 모델 이용하여 이미지 인코딩 프로세스를 되돌아가며 학습한다. 여기서 Diffusion 모델을 사용한다. Prior 모델 이용하여 이미지 캡션의 텍스트 인코딩을 해당 이미지의 이미지 인코딩으로 매핑한다. 이를 통합한 후, CLIP 텍스트 인코더가 이미지 설명을 표현 공간에 매핑한다. DIffusion Prior가 CLIP 텍스트 인코딩을 관련 CLIP 이미지 인코딩으로 매핑한다. 수정된 GLIDE 생성 모델이 역확산을 이용해 표현 공간을 이미지 공간으로 매핑하고, 입력된 캡션 내에서 Semantics를 전달하는 이미지들을 생성하게 된다.

## 2.3 Diffusion Model

Diffusion 모델은 Diffusion process와 Reverse Diffusion process로 이루어져 있다. 먼저 Diffusion process는 주어진 데이터  $x$ 에 점점 noise를 추가한다. 이 과정은 주어진 데이터의 분포를 파괴하는 단계이다. 그 다음 Reverse Diffusion process에서는 앞에서 정의한 Diffusion process의 역과정을 계산한다. 즉 이 단계에서 원래의 데이터인  $x$ 로 돌아오게 된다. Diffusion process는 직접 Gaussian distribution으로 정의했지만 반대 방향인 Reverse Diffusion process를 표현하는 distribution들은 모른다. 따라서 모델을 통해 모르는 Reverse Diffusion process를 학습한다. 이를 통해 noise data에서 점점 noise를 제거할 수 있다. Diffusion model은 복잡하고 고차원인 이미지 데이터의 분포를 모사하고, 분포로부터 이미지를 샘플링 할 수 있다. 이미지 생성 과정에서 condition을 주고 생성을 하는 경우 guided diffusion이라 한다. Condition으로 class 정보를 줄 수도 있고, text description을 줄 수도 있다.

## 2.4 ARKit

ARKit은 iOS 증강 현실(Augmented Reality) 애플리케이션을 구축하기 위해 애플에서 개발한 소프트웨어 프레임워크이다. ARKit은 장치의 카메라와 센서를 사용하여 환경을 감지하고 장치의 위치와 방향을 실시간으로 추적한다. 이를 통해 사용자의 움직임과 제스처에 반응하고 상호 작용하는 AR 도구를 만들 수 있다. ARKit은 테이블이나 바닥과 같은 실제 표면에 가상 개체를 고정할 수 있는 평면 감지와 같은 여러 기능을 포함한다. 또한 환경에 따라 정확한 양의 빛으로 가상 물체를 비출 수 있는 빛 추정을 지원한다.

## 2.5 VisionKit

VisionKit은 애플에서 개발한 iOS 머신 러닝 및 컴퓨터 비전 프레임워크이다. iOS 디바이스에서 사용할 수 있으며 머신 러닝을 활용하여 이미지와 텍스트를 인식하고 해석할 수 있다. VisionKit은 앱이 실시간으로 이미지에서 텍스트를 감지하고 추출할 수 있도록 하는 텍스트 인식(OCR)과 같은 여러 기능을 포함한다. 또한 이미지와 동영상에서 얼굴을 식별하고 추적하는 데 사용할 수 있는 얼굴 감지 기능과 특정 개체를 실시간으로 추적할 수 있는 개체 추적 기능이 포함되어 있다. 고급 기계 학습 알고리즘을 사용하여 높은 수준의 정확성과 성능을 가지고 이미지와 텍스트를 분석하기 때문에 빠르고 정확한 결과를 제공한다. VisionKit은 사용하기 쉽고 기존 iOS 애플리케이션에 통합되어 있다. 기능에 액세스하

는 데 사용할 수 있는 일련의 API와 얼굴 또는 텍스트와 같은 특정 유형의 콘텐츠를 인식하고 해석하는 데 사용할 수 있는 사전 훈련된 모델 범위를 제공한다.

### 3. 프로젝트 내용

#### 3.1 문제 정의

Stable Diffusion 모델을 활용하여 객체를 생성하는 모바일 어플리케이션을 구현한다. OCR을 사용한 키워드 자동 인식, 직접 검색 등 다양한 방식으로 사용자가 찾는 키워드를 검색할 수 있도록 구현하고, 현재 진행 중인 독서를 기록하고 키워드 조회 기록을 저장하는 기능을 구현한다.

#### 3.2 시나리오

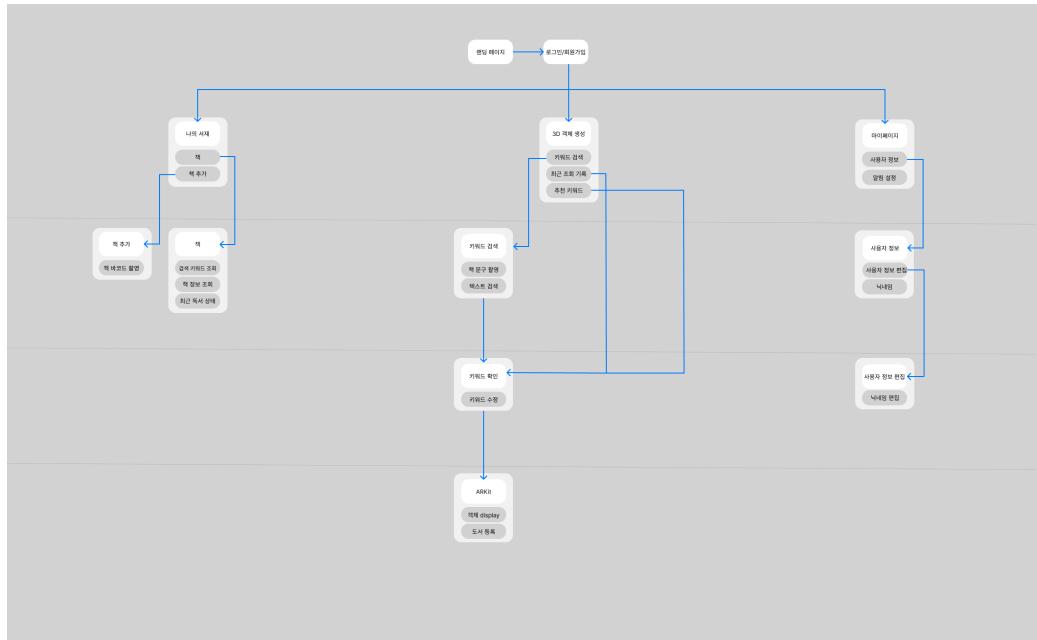
- A. 사용자는 책이나 특정 텍스트에서 찾은 객체를 설명하는 부분을 카메라나 사진을 통해 모바일 어플리케이션에 업로드 한다.
- B. OCR 기능을 통해 사진에서 텍스트를 추출한다. 이후 사용자에게 확인 또는 수정을 거친다.
- C. 추출한 텍스트를 번역 모듈을 사용하여 영어로 번역한다.
- D. 영어로 번역된 텍스트를 Stable Diffusion 모델을 사용하여 이미지를 생성한다.
- E. 생성된 이미지를 ARKit을 사용하여 시각화한다.

#### 3.3 모듈 요구 사항

OCR 모듈	번역 모듈
VisionKit	Ko to En
Image to Text	Papago Translator
생성 모델 데이터 처리 모듈	객체 시각화 모듈
Stable Diffusion	ARKit
이미지 처리	객체 생성 + 이미지 추가

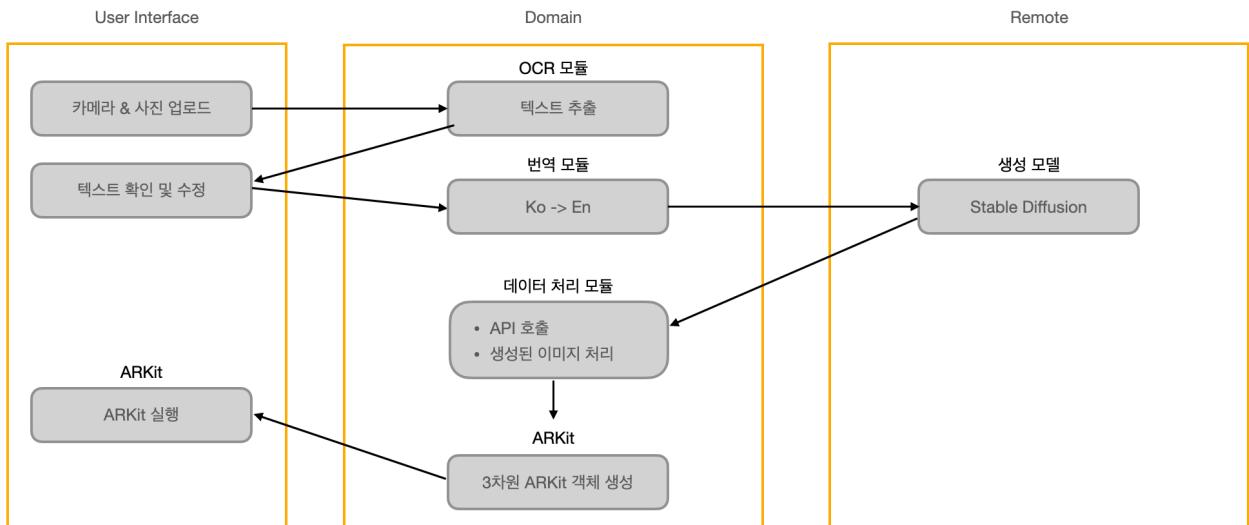
[표 1] 요구 모듈 명세서

### 3.4 System Architecture



[그림 2] Information Architecture 설계

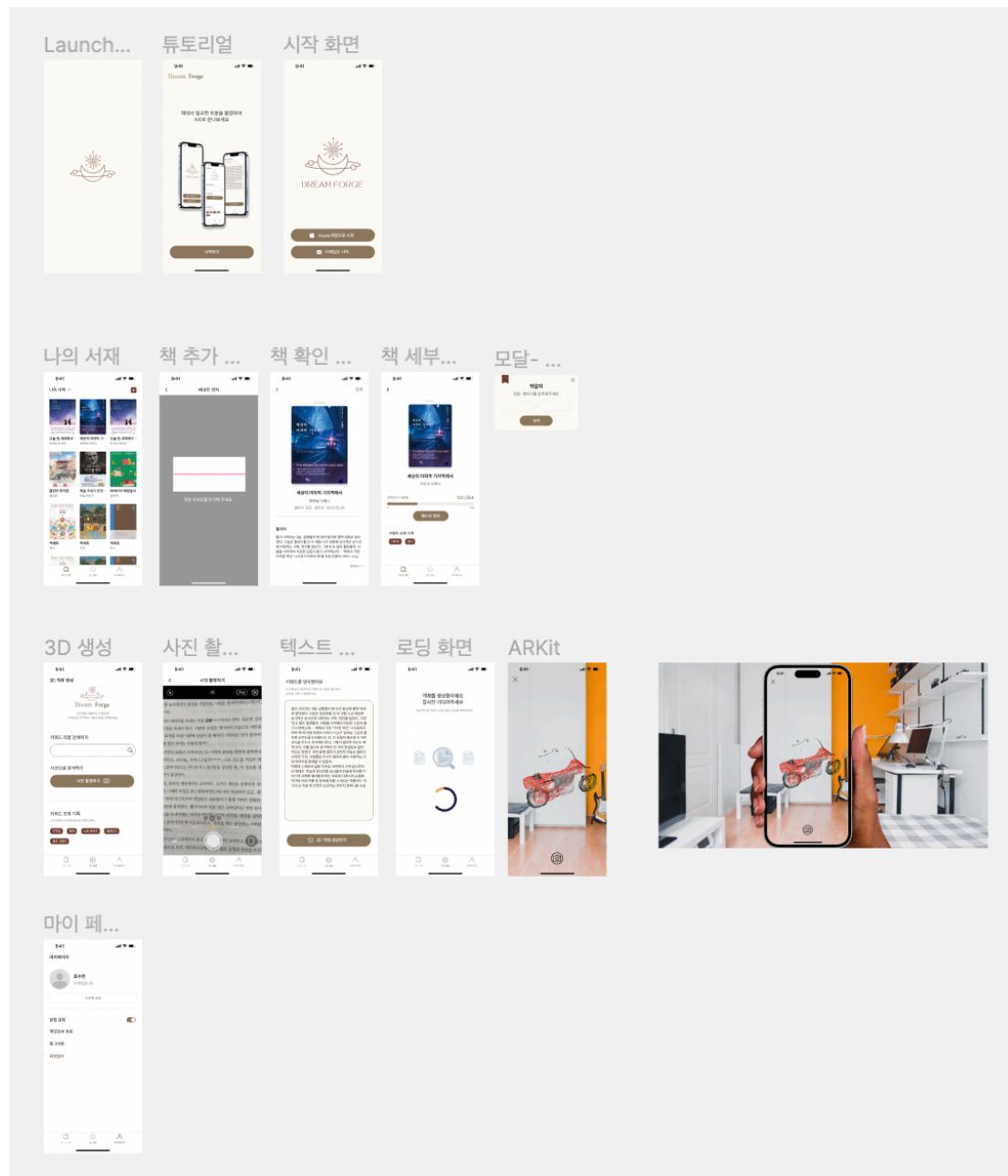
Information Architecture는 [그림 1]과 같다. 사용자의 독서 상황과 그에 따른 키워드 검색 내역을 저장하고 확인할 수 있는 ‘나의 서재’, 카메라 또는 직접 검색을 통해 키워드를 사용하여 3D 객체를 생성하는 ‘3D 생성’, 사용자의 정보를 설정하고 관리하는 ‘나의 페이지’로 구성된다. 이를 위한 시스템 아키텍처는 다음 [그림 2]와 같다.



[그림 3] System Architecture Overview

시스템은 크게 4개의 모듈을 함께 사용한다. 먼저 사용자가 카메라를 사용하여 키워드를 검색할 때, 사진에서 텍스트를 가져오기 위한 OCR 모듈을 구현한다. 가져온 텍스트를 사용자가 편집하여 키워드를 추출한다. 이때 Stable Diffusion에 키워드 또는 텍스트를 적용하기 위해서 번역 모듈을 구현하여 영어로 번역할 수 있도록 설계한다. Stable Diffusion 모델의 결과로 나온 이미지를 사용하여 ARKit 상에서 객체로 나타내는 객체 표현 모듈을 구현하여 최종적으로 사용자에게 객체를 보여준다.

이를 바탕으로 GUI 디자인을 설계한다. 디자인 툴은 Figma를 사용하였다.



[그림 4] GUI 디자인

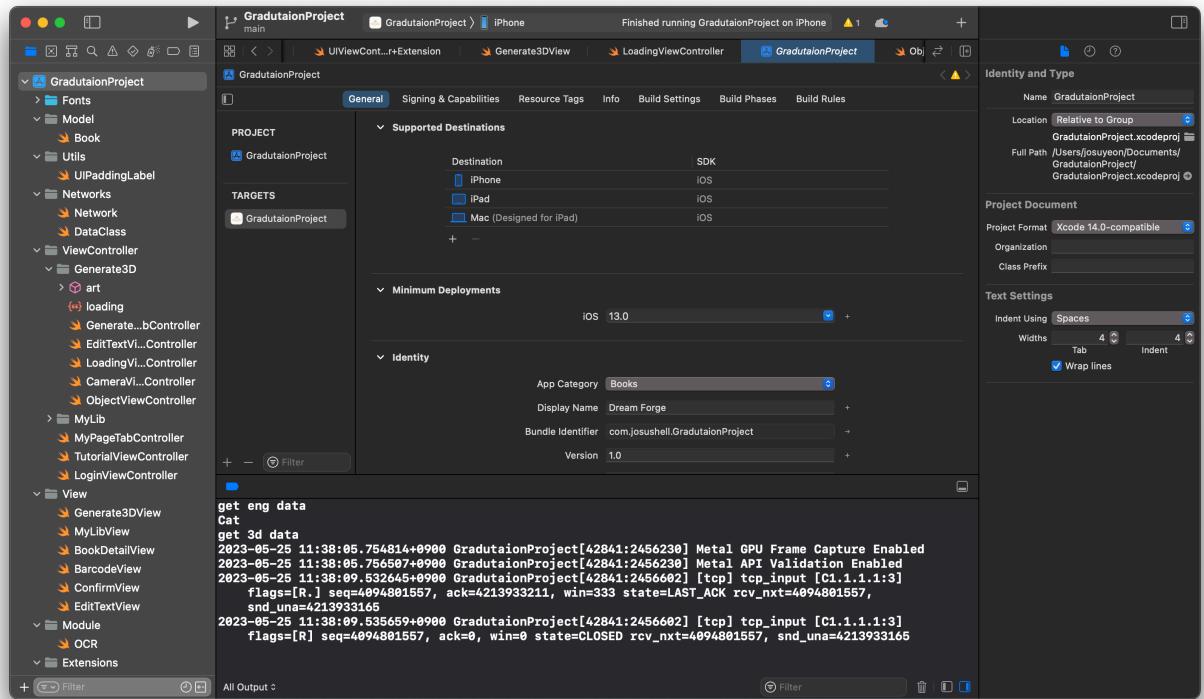
## 3.5 시스템 구현

### 3.5.1 개발 환경

개발 환경은 다음과 같이 구성한다.

개발 툴	Xcode 14.2
최소 빌드 버전	iOS 13
개발 언어	Swift 4.0

[표 2] 개발 환경 구성



[그림 4] Xcode 환경 세팅

### 3.5.2 OCR 모듈 코드 설계

이미지에서 텍스트를 추출하는 OCR 모듈은 VisionKit을 사용하였다.

VNRecognizeTextRequest 클래스를 사용하여 텍스트를 인식한다. 기본적으로 텍스트 인식 요청은 먼저 입력 이미지에서 가능한 모든 glyphs 또는 문자를 찾은 다음 각 문자열을 분석한다. 이때 recognitionLanguages 속성을 통해 한국어를 인식하도록 설정한다. 인식된 텍스트는 VNRecognizedTextObservation 객체로 반환되도록 설계한다.

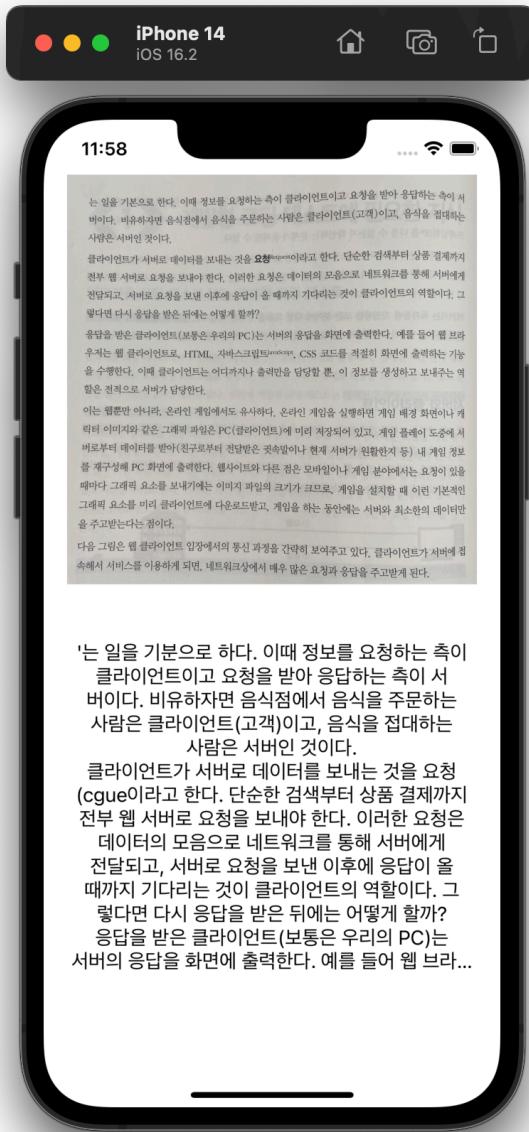
```

7
8 import UIKit
9 import Vision
10 import VisionKit
11
12 class OCR {
13     func recognizeText(image: UIImage?, completion: @escaping ((String) -> Void)){
14         guard let cgImage = image?.cgImage else {
15             fatalError("could not get image")
16         }
17
18         let handler = VNImageRequestHandler(cgImage: cgImage, options: [:])
19         let request = VNRecognizeTextRequest{ request, error in
20
21             guard let observations = request.results as? [VNRecognizedTextObservation],
22                 error == nil else{
23                 return
24             }
25
26             let text = observations.compactMap({
27                 $0.topCandidates(1).first?.string
28             }).joined(separator: "\n")
29
30             DispatchQueue.main.async {
31                 completion(text)
32             }
33         }
34
35         if #available(iOS 16.0, *) {
36             let revision3 = VNRecognizeTextRequestRevision3
37             request.revision = revision3
38             request.recognitionLevel = .accurate
39             request.recognitionLanguages = ["ko-KR"]
40             request.usesLanguageCorrection = true
41
42             do {
43                 var possibleLanguages: Array<String> = []
44                 possibleLanguages = try request.supportedRecognitionLanguages()
45                 print(possibleLanguages)
46             } catch {
47                 print("Error getting the supported languages.")
48             }
49         } else {
50             // Fallback on earlier versions
51             request.recognitionLanguages = ["en-US"]
52             request.usesLanguageCorrection = true
53         }
54
55         do{
56             try handler.perform([request])
57         } catch {
58             print(error)
59         }
60     }
61 }

```

[그림 5] OCR 모듈 코드

OCR 모듈을 simulator로 직접 테스트한 결과는 다음 [그림 6]과 같다.



[그림 6] OCR 모듈 테스트 결과

이때 글자가 사진에 나오지 않아 잘리는 경우를 제외하고는 모두 정확하게 텍스트가 인식되었다.

### 3.5.3 번역 모듈 코드 설계

번역기는 Papago API를 사용하였다. 텍스트를 인식하는 경우 인식된 텍스트를 영어로 번역하기 위해 API에 접근한다. 이때 RxSwift를 사용하여 네트워크 접근을 비동기로 처리한다. 번역된 텍스트가 온 경우 이를 Stable Diffusion 모델에 전달한다.

```

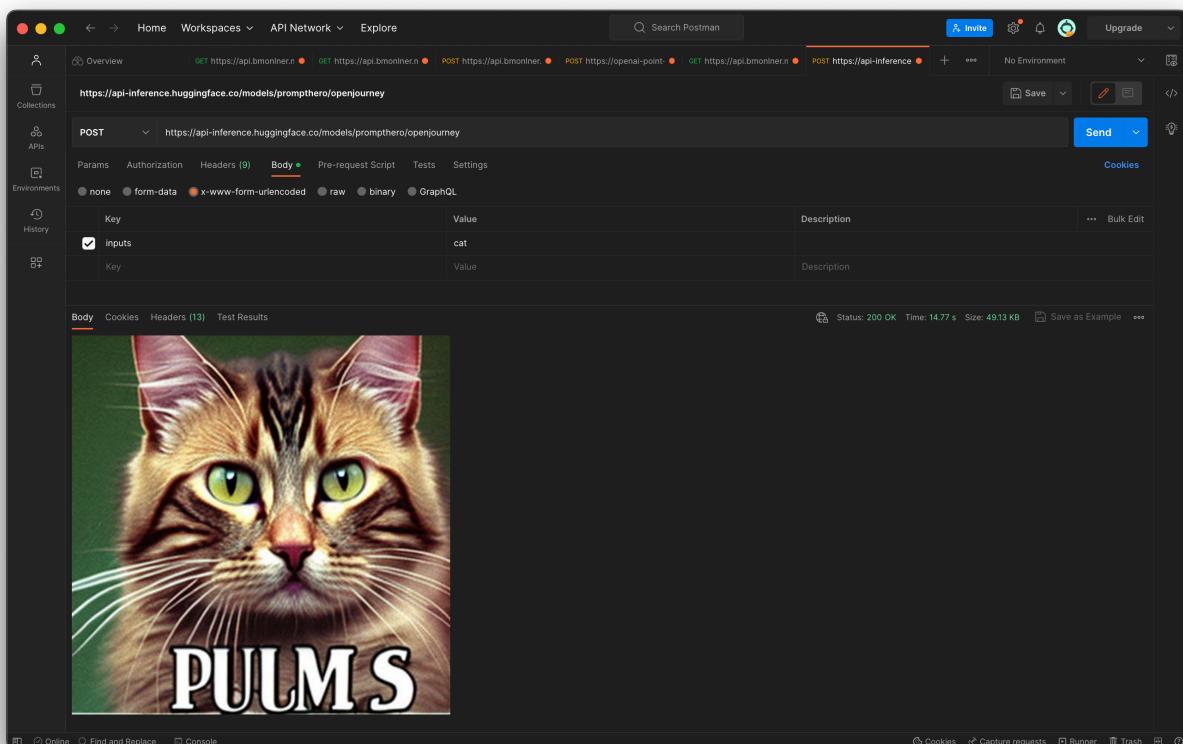
56     func translateKoTOEn(text: String) -> Observable<String> {
57         return Observable<String>.create { observer in
58             let url = "https://openapi.naver.com/v1/papago/n2mt"
59             let params = ["source":"ko",
60                         "target":"en",
61                         "text": text]
62             let header: HTTPHeaders = ["Content-Type":"application/x-www-form-urlencoded; charset=UTF-8",
63                                         "X-Naver-Client-Id":"NiY4h3mNnqwmdm_rH0qR",
64                                         "X-Naver-Client-Secret":"YPcCctLA5f"]
65
66             AF.request(url, method: .post, parameters: params, encoding: URLEncoding.default,
67                        headers: header).responseDecodable(of: Translator.self, completionHandler: { response in
68                 switch response.result {
69                     case .success(let data):
70                         observer.onNext(data.message.result.translatedText)
71                         observer.onCompleted()
72                     case .failure(let e):
73                         print(e)
74                         observer.onError(e)
75                 }
76             })
77
78             return Disposables.create {}
79         }.asObservable()
80     }
81 }
82
83

```

[그림 7] 번역 모듈 코드

### 3.5.4 생성 모델 데이터 처리 모듈

Stable Diffusion에 번역된 텍스트를 넣어 이미지를 생성하고 결과를 받아오는 모듈이다. 이때 Prompt를 통해 제공되는 API를 사용하여 비동기로 이미지를 생성하고 처리한다. 테스트 결과는 다음 [그림 8]과 같다.



[그림 8] Stable Diffusion 모델 테스트

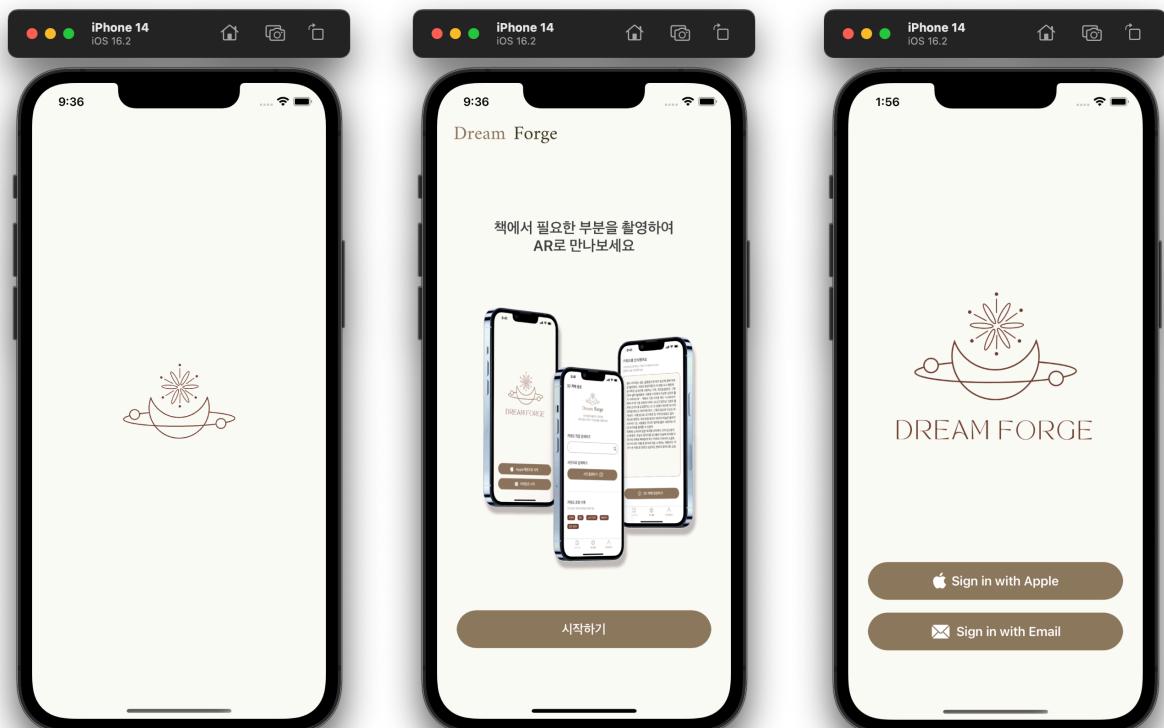
는 다음 [그림 8]과 같다. 이때 생성된 이미지는 바이너리 Data의 형태로 전달된다. 따라서 UIImage 클래스를 통해서 인식 가능한 이미지 형태로 변환한다.

```
12 class Network {
13     func getOpenJourneyModel(keyword: String) -> Observable<UIImage?> {
14         return Observable<UIImage?>.create { observer in
15             let url = "https://api-inference.huggingface.co/models/prompthero/openjourney"
16             let param = ["inputs" : keyword]
17             let header : HTTPHeaders = [
18                 "Authorization" : "Bearer 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"
19             ]
20             AF.request(url, method: .post, parameters: param, encoding: URLEncoding.default, headers:
21                         header).responseData { response in
22                 switch response.result {
23                     case .success(let img):
24                         observer.onNext(UIImage(data: img))
25                         observer.onCompleted()
26                     case .failure(let e):
27                         print(e)
28                         observer.onError(e)
29                 }
30             }
31             return Disposables.create {}
32         }.asObservable()
33     }
34 }
```

[그림 9] 생성 모델 데이터 처리 모듈 코드

## 4. 프로젝트 결과

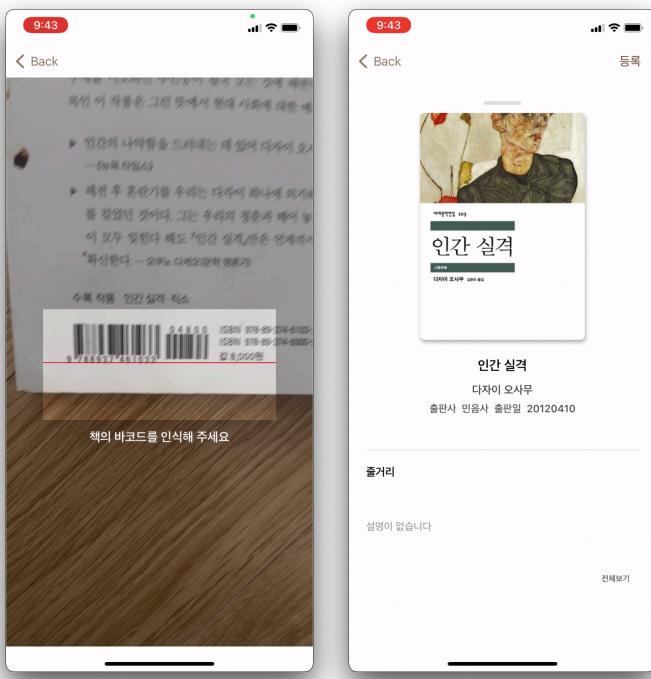
### 4.1 스플래시 및 회원가입



[그림 10] 스플래시 및 회원 가입 화면

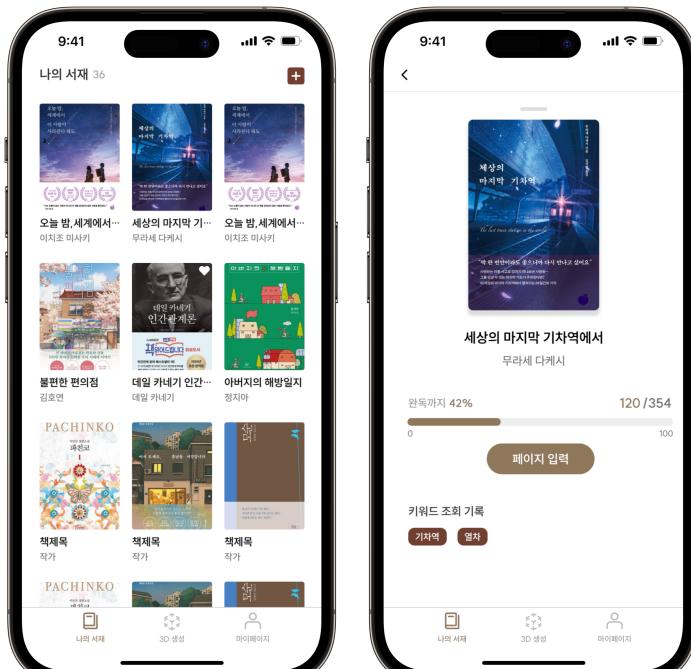
## 4.2 나의 서재 기능

사용자는 스마트폰 카메라를 사용하여 실제 도서의 바코드를 촬영하여 isbn을 인식하거나 사용자가 직접 isbn을 검색한다. isbn을 가져온 후 네이버 책 검색 API를 사용하여 세부 책 정보를 가져온 다음 서재에 등록할 수 있다.



[그림 11] 바코드 인식 기능

서재에 있는 도서를 터치하면 사용자는 독서 진행 상황을 기록하거나 키워드를 검색한 이력을 조회할 수 있다.

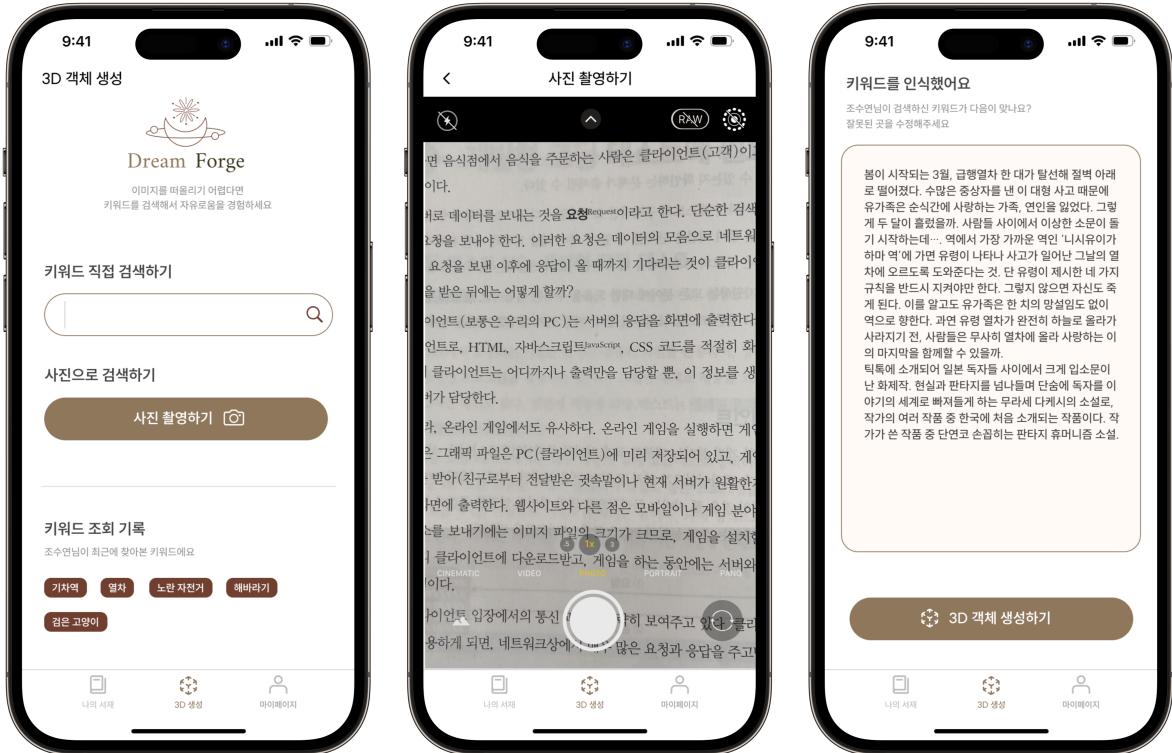


[그림 12] 나의 서재 기능

#### 4.3 이미지 생성 기능

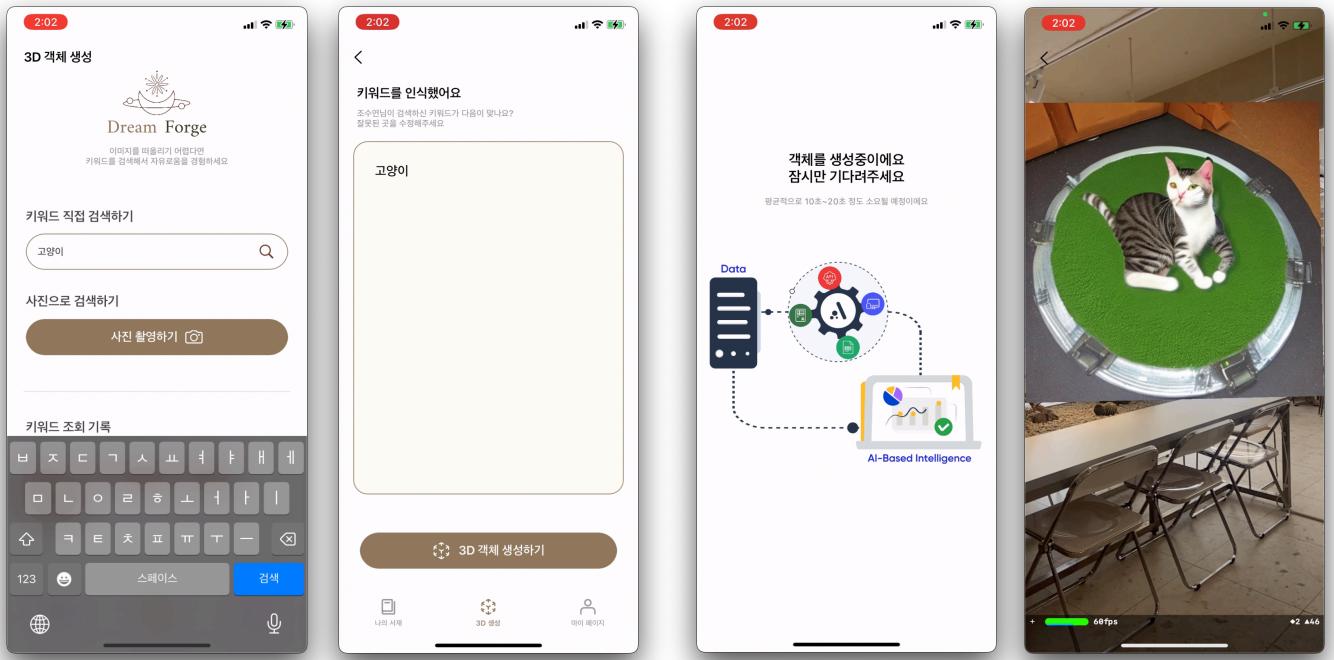
사용자는 키워드를 직접 검색하거나 nvbc나 스마트폰 카메라로 도서를 촬영하여 검색하고자 하는 텍스트를 추출할 수 있다. 이때 촬영된 사진에서 Apple VisionKit을 사용하여 구현해둔 OCR 모듈을 통해 텍스트를 인식한다. 이후 인식된 텍스트에서 사용자가 원하는 부분을 수정할 수 있도록 수정 기능을 제공한다. 추출한 키워드의 언어를 감지하여 한글인 경우 네이버 번역기 API를 사용하여 영어로 키워드를 번역한다.

이후 Stable Diffusion 모델의 prompt에 키워드를 넣어 이미지를 생성한다. 생성 모델 데이터 처리 모듈을 사용하여 반환된 UIImage 객체를 가지고 ARKit Scene에 나타낸다. 이때 SCNBox 객체를 사용하여 Scene 상에서 표현한다.



[그림 13] 카메라와 OCR 모듈을 사용한 키워드 추출

[그림 14]와 같이 직접 검색을 통해서도 이미지를 생성할 수 있도록 구현한다. 키워드를 검색하고 AR Scene에서 보여주는 과정은 위와 동일하다.



[그림 14] 직접 검색을 통한 이미지 생성

## 5. 결론 및 기대 효과

아판타시아 증후군을 가진 사람들이 독서를 할 때 특정 객체를 시각화하지 못하는 문제를 해결하기 위해, text-to-image 생성 모델을 활용한 모바일 어플리케이션을 구현한다. 이를 통해 환자는 일상 생활에 긍정적인 영향을 미칠 수 있는 시각화 기술을 연습하고 향상시킬 수 있다. 또한 환자에게 연습할 수 있는 시각화를 제공함으로써 인지 재활을 도울 수 있다. 시각화는 기억 형성과 회상에 중요한 역할을 한다. 따라서 환자가 특정 기억과 연관시킬 수 있는 시각화를 제공하여 환자의 기억력을 향상시키는 데 도움을 줄 수 있다. 또한 아판타시아 증후군을 가진 환자들에게 독서에 몰입할 수 있는 기회를 제공하고 상상력을 증진시켜 증후군을 극복하고 일상에 도움을 줄 수 있다.

## 6. 참고 문헌

- [1] 한국과학기술원(2022) 「최신 Text-to-Image 생성 모델의 동향」
- [2] 박하나(2022) 「이미지 생성 인공지능(AI) 달리(DALL-E)의 활용 사례 연구」

[3] Aditya Ramesh, Mikhail Pavlov (2021) 「DALL-E: Zero-Shot Text-to-Image Generation」