“Make it Move”



|  |  |
| --- | --- |
| 과목명 | 모바일앱프로그래밍1 |
| 담 당 | 정 기 숙 교수 |
| 작성일 | 2023년 12월 20일 |
| 프로젝트명/조 | MiM/9조 |
| 조 원 | 김서진 (2020112348) |
|  | 김호준 (2019113169) |
|  | 박주용 (2021428763) |
|  | 가파로브 자혼기르(2021111835) |

목차

**1. 프로젝트 소개**

A. 개요

B. 특징

C. 기대효과

**2. 사용자 매뉴얼 및 기능 설명**

A. 메인 화면

B. 이미지 선택

C. 동영상 선택

D. 다운로드 결정

**3. 기술 설명**

A. 공통 함수

1. ApiService 클래스

2. Ext 클래스

3. FormDataUtil 클래스

4. FileUtils 클래스

5. NetWorkModule 클래스

B. Activity클래스

**4. 프로젝트 진행 방식**

A. 일정

B. 역할분담

**5. 개선사항**

**1.** **프로젝트 소개**

**A. 개요**

“Make it Move”는 이미지에 대한 움직임을 생성하는 딥러닝 모델을 구체화시킨 어플리케이션 중 하나이며 사용자 친화적인 UI와 쉬운 조작, 가벼운 어플리케이션을 통해 사용자에게 AI와 연계된 경험을 제공합니다.

**B. 특징**

**- 사용자 친화적 UI:** “Make it Move”내 사용자와 상호작용하는 모든 View와 Button은 사용자가 그 기능과 목적을 쉽게 식별할 수 있도록 설계되었으며 이는 사용자에게 친숙하고 편안한 경험을 제공할 수 있도록 도움을 줍니다.

**-** **간편한 조작:** 필요성이 다소 떨어지거나 복잡한 레이아웃을 배제하여 어떤 사용자라도 쉽게 사용할 수 있도록 도움을 줍니다.

**-** **이미지 편집 기능:** 이미지를 영상으로 변환하도록 서버로 요청하기 전 성공적인 변환을 위해 사전에 이미지를 편집할 수 있도록 하여 불필요한 자원 소모를 줄입니다.

**- 빠른 동작 속도:** 이미지 편집 기능으로 인한 불필요한 자원 소모 감소와 함께 서버에서 작업량이 큰 요청은 별도로 분류에 다수의 사용자에게 만족스러운 경험을 제공할 수 있도록 합니다.

**C. 기대효과**

**1. AI 사용 경험 제공**

전문적인 지식과 자원을 요구하는 AI 사용 경험을 “Make it Move”를 통해 무료로 손쉽게 경험해볼 수 있습니다.

**2. 여러 모델을 결합한 어플리케이션 플랫폼 조성**

현재 적용 중인 모델 뿐만 아니라 다른 최신 모델을 구체화하여 사용자에게 제공함으로써 이미지 변환 어플리케이션이 아닌 AI 체험 플랫폼으로 거듭날 발판을 마련할 수 있습니다.

**2. 사용자 매뉴얼 및 기능 설명**

**A. 메인 화면**

**‘**나만의 애니메이션 만들기’ Button을 이용해 다음 단계로 진입할 수 있습니다.

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

**B. 이미지 선택**

사용자가 변환을 요청할 이미지를 갤러리에서 선택하거나 카메라를 이용하여 직접 촬영할 수 있도록 하였고 선택/촬영한 이미지는 즉시 편집할 수 있습니다.

텍스트, 스크린샷, 직사각형이(가) 표시된 사진

자동 생성된 설명 인간의 얼굴, 스크린샷, 사람, 텍스트이(가) 표시된 사진

자동 생성된 설명 인간의 얼굴, 텍스트, 스크린샷, 사람이(가) 표시된 사진

자동 생성된 설명

**C. 동영상 선택**

이미지가 움직이는 기준이 될 동영상을 선택할 수 있도록 하며 동시에 이전 단계에서 편집한 이미지를 표시해 착오를 방지합니다.

텍스트, 인간의 얼굴, 사람, 스크린샷이(가) 표시된 사진

자동 생성된 설명

**D. 다운로드 결정**

이미지와 동영상이 선택되면 즉시 서버에 변환을 요청하며 변환이 종료되면 사용자에게 다운로드 여부를 결정하는 Fragment가 출력됩니다.

텍스트, 인간의 얼굴, 스크린샷, 사람이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 인간의 얼굴이(가) 표시된 사진

자동 생성된 설명

**3. 기술 설명**

- 이미지 변환 단계에 따른 4개의 Activity(Fragment)와 5개의 클래스에 대한 함수 설명.

**A. 공통 함수**

**1. ApiService 클래스**

* suspend fun postImageAndVideo(..): 선택된 사진과 영상을 Multipart/form-data형식으로 서버에 요청하는 함수, Response가 Json 형식이 아닌 text파일로 오기 때문에 Response<ResponseBody>를 반환한다.
* suspend fun getAnimation(..): 서버로부터 변환된 영상을 다운로드하기 위해 호출하는 함수, postImageAndVideo()로부터 응답받은 String타입의 ticketID를 RequestBody로 설정하여 요청한다.

**2. Ext 클래스**

* Int.checkIsUpperSdkVersion(): 현재 기기의 SDK\_VERSION을 확인하는 함수, VERSION에 따라 미디어 파일 접근시 필요한 권한이 다르게 요구되기 때문에 체크가 필요하다.
* Activity.getDeviceSize(): 현재 기기의 ScreenSize를 구하는 함수, VERSION에 따라 다르게 처리한다.

**3. FormDataUtil 클래스**

* Uri.asMultipart(..): 갤러리나 카메라로부터 가져온 사진과 영상들을 MultipartBody.Part 타입으로 변환하는 함수.

**4. FileUtils 클래스**

* fun createTempImageFile(..): 카메라로 사진을 촬영 후, 해당 사진을 저장할 파일을 생성하는 함수.
* fun filePathToUri(..): Image, Video의 각 경로를 파라미터로 받아 Uri로 변환하는 함수.
* fun isFileInInternalStorage(..): 파일 경로가 내부 저장장치인지 외부 저장장치인지 판단하는 함수.

**5. NetWorkModule 클래스**

* 통신에 필요한 OKHttpClient, Retrofit 인스턴스가 정의되어 있다.

**B. Activity 함수**

**1. MainActivity.kt**

* fun onCreate(..): buttonStart를 통해 다음 단계로 이동하기 위한 Intent를 생성한다.

**2. SelectImageActivity.kt**

* private val fetchPicturesFromGallery = registerForActivityResult(..): 갤러리에서 이미지를 선택하면 ActivityResult가 반환되며 정상적으로 Uri가 담겨져서 반환되면 ImageCrop을 진행한다.
* private val cameraActivityLauncher = registerForActivityResult(..): 카메라 촬영이 정상적으로 수행되었는지 여부가 Boolean값으로 변환된다, 정상적으로 Uri가 반환되면 ImageCrop을 진행한다.
* private fun takePicture(): 사진 촬영을 수행하는 함수, 촬영이 성공적으로 완료되면 해당 사진을 저장한 File 인스턴스를 생성하고 ActivityResultLauncher를 이용해 카메라를 실행한다.
* private val requestPermissionLauncher = registerForActivityResult(..): 미디어 파일 권한 체크를 담당하며 권한이 모두 허용되어 있다면 사용자의 요청에 따라 갤러리로 이동하거나 카메라 촬영을 수행한다.
* private val cropImage = requestForActivityResult(..):, Crop된 이미지의 Uri를 결과로 반환받는다.
* private fun setImageUri(..): ImageView에 특정 이미지의 Uri를 전달받아 화면에 표시하도록 하는 함수.
* private fun startCrop(..): 갤러리에서 선택하거나 카메라로 촬영한 파일의 Uri를 파라미터로 받은 후 ImageCrop ActivityResultLauncher를 실행하는 함수.

**3. SelectVideoActivity.kt**

* private fun getSelectedImage(): 이전 단계인 SelectImageActivity.kt에서 선택된 이미지의 Uri를 Intent를 통해 전달받은 후 화면에 출력시키는 함수.
* Private fun selectVideo(): 동영상 선택 버튼을 권한 체크와 연계시키는 함수.
* Private val requestGalleryVideoPermission = registerForActivityResult(..): 갤러리에서 비디오를 선택하기 위한 권한을 체크하고 필요한 모든 권한이 충족되었다면 Intent를 이용해 갤러리에서 비디오를 선택할 수 있도록 한다.
* Private val fetchPicturesFromGallery = registerForActivityResult(..): 갤러리에서 비디오가 선태되면 ActivityResult를 반환받아 Uri를 이용해 bitmap을 추출하고 Glide 라이브러리를 활용해여 ImageView에 선택된 동영상의 썸네일 이미지를 출력시킨다.
* Private fun getVideoThumbnail(..): 선택된 Video Uri에서 썸네일 이미지를 Bitmap 형식으로 추출하여 반환하는 함수, 적절한 프레임을 지정하여 원하는 위치의 썸네일을 추출할 수 있다.
* Private fun uploadMediaFile(): 이미지와 동영상 Uri를 MultpartBody.Par로 변환 후, Retrofit을 이용하여 서버에 POST하는 함수, Http 통신이 성공적으로 완료되면 ResponseBody에 ticket 정보가 반환된다.
* Private fun getAnimation(): 서버로부터 반환된 ticket을 이용해 변환된 애니메이션을 반환받기 위한 함수, 변환이 완료되지 않으면 Response의 Content-Type을 application/json, 변환이 완료되었다면 video/mp4로 응답받는다.
* Private fun natToMain(): 다운로드 완료 후 메인화면으로 돌아가는 함수.
* Private suspend fun showLoading(): 네트워크 통신 여부에 따라 ProgressBar의 Visibility를 변경하는 함수, Http 통신은 대부분 IO Dispathers에서 일어나기 때문에 Dispathers.Main으로 전환하여 작업을 수행한다.
* Private suspend fun showLoadingMessage(): 네트워크 통신 여부에 따라 메시지를 변경시키는 함수.
* Private fun checkDirectoryExist(): 파일을 쓰기 전에 생성하고자 하는 경로가 유효한지 확인하는 함수, 경로가 유효하지 않다면 폴더를 생성한다.
* Private fun writeToDisk(): 서버로부터 응답받은 ResponseBody를 전달받아 갤러리에 저장하는 함수, FileOutputStream을 활용한다.

**4. DownloadDialogFragment.kt**

* Override fun onViewCreated(..): 다운로드 버튼 클릭시 DownloadListener를 통해 SelectVideoActivity.kt로 결과를 반환한다.
* Private fun setDialogSize(): Dialog의 Size를 지정하는 함수, getDeviceSize()를 통해 실제 기기의 width와 height를 구한 후, 원하는 비율로 다이얼로그의 크기를 설정한다.

**4. 프로젝트 진행 방식**

**A. 일정**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **10/17** | **11/8** | **11/15** | **11/22** | **11/29** | **12/7** | **12/14** | **12/21** |
| **주제선정 및 제안서 작성** |  |  |  |  |  |  |  |  |
| **역할 분담** |  |  |  |  |  |  |  |  |
| **백엔드 개발** |  |  |  |  |  |  |  |  |
| **어플리케이션 개발** |  |  |  |  |  |  |  |  |
| **기술문서 작성** |  |  |  |  |  |  |  |  |
| **발표 준비** |  |  |  |  |  |  |  |  |

**B. 역할 분담**

**1. 김서진**

**- 주요 역할:** 어플리케이션(Class)

**2. 김호준**

**- 주요 역할:** 어플리케이션(Class, Layout), PPT 제작

3. **박주용**

**- 주요 역할**: 백엔드 개발, 어플리케이션(C-S Communication), 문서 작성

4. **가파로브 자혼기르**

**- 주요 역할**: 어플리케이션(Class), 발표

**5. 개선사항**

**A. 이미지-비디오 변환 성공률 향상**

**-** 항상 사용자가 입력하는 데이터를 만족스럽게 변환할 수 없음에도 불구하고 서버의 자원을 소비하는 문제가 있으며 어플리케이션 단에서 미리 적절한 전처리 후 전송하도록 유도해야 함.

**B. 커스터마이징 제공**

**-** 어플리케이션 단에서 변환을 수행할 모델을 직접 지정할 수 있도록 해 만족스러운 경험을 제공하도록 개선해야 함.