**Image-Searching UI 2차 설계서**

@@@ 전체적 시스템 구조 및 UI 디자인 @@@

Image-Searching Layout이 제공하는 기능에는 이미지 검색 기능과 일기 작성 기능 및 사용자가 작성해 둔 일기 내용에서 키워드를 추출하여 태그를 추가하는 기능이 있다. 이미지 검색 기능은 태그 정보 및 촬영 날짜와 촬영 장소 정보를 저장하는 데이터베이스를 구현한 후, 사용자가 태그 정보, 촬영한 날짜(기간으로 입력) 및 촬영 장소 정보를 입력했을 때 입력한 정보와 일치하는 이미지들을 데이터베이스에서 찾아 사용자에게 해당 이미지의 목록을 보여주는 구조로 구현할 예정이다. 또한 일기를 작성한 후 키워드를 선택하여 태그를 추가하는 기능은 사용자가 일기 내용에서 키워드를 선택했을 때 해당 이름을 가진 태그를 데이터베이스에 추가로 저장하는 구조로 구현하고자 한다.

도표이(가) 표시된 사진

자동 생성된 설명

\*\*\* 이미지 검색 화면 \*\*\*

텍스트이(가) 표시된 사진

자동 생성된 설명

사용자가 입력창에 태그 이름을 입력하거나, YYYY-MM-DD의 형식으로 기간을 지정하거나, Google Maps API를 이용하여 장소 정보를 입력할 경우 입력 정보와 일치하는 이미지들의 목록이 ImageView에 자동으로 나타나며, 좌우 이동 버튼을 누르면 이전 이미지 또는 다음 이미지로 이동할 수 있다. 이 때 Google Map을 이용하여 촬영 장소로 이미지를 검색하는 기능은 사용자가 Google Map 내에서 위치를 선택했을 때, 이미지 파일 내에 있는 GPS 좌표 정보와 사용자가 선택한 장소의 좌표 값을 이용하여 주소 정보를 추출한 후, “OO도 OO시 OO구/군”까지의 정보가 일치하는 이미지를 검색하도록 할 것이다. 또한, UI 하단에 존재하는 “Write a Diary” 버튼을 클릭했을 경우에는 선택한 이미지에 대해 일기를 작성하는 화면으로 넘어간다.

\*\*\* 일기 작성 화면 \*\*\*

차트이(가) 표시된 사진

자동 생성된 설명

UI의 좌측 상단에는 사용자가 선택한 이미지를 보여 줄 것이며, store 버튼을 클릭할 경우 사용자가 작성한 일기를 저장할 수 있고, modify 버튼을 누르면 작성했던 일기의 내용을 수정할 수 있으며, delete 버튼을 누르면 사용자가 작성했던 일기를 삭제할 수 있다. 또한 “select keyword” 텍스트의 오른쪽에는 체크 박스가 있는데, 이 체크 박스에 체크가 되어 있지 않다면 UI는 일기의 저장/수정/삭제 기능을 제공하며, 체크가 되어 있다면 사용자가 일기 내용에서 키워드를 몇 개 선택한 후 선택한 키워드를 이용하여 태그를 추가적으로 생성하는 기능을 제공하게 된다.

@@@ 사용 기술 @@@

* Google Maps API: 촬영 장소를 이용하여 이미지를 찾는 기능을 구현할 때 사용 예정
* SQLite: 안드로이드에서 제공하는 경량 데이터베이스
* ContentProvider: 데이터를 안드로이드 애플리케이션 간에 공유하기 위한 인터페이스
* CurserLoader: 데이터를 비동기적으로 로드하고 화면에 표시하기 위한 유틸리티 클래스

@@@ 함수 정의 및 설명 @@@

**Gallery - 데이터베이스 간 연동**

galleryInfoLink()

갤러리에 존재하는 사진들의 정보를 연동하는 함수

**이미지 검색 화면**

onPrevButtonPressed()

이미지 검색 화면에서 이전 이미지로 이동하는 버튼을 클릭했을 때의 이벤트를 처리하는 함수

onNextButtonPressed()

이미지 검색 화면에서 다음 이미지로 이동하는 버튼을 클릭했을 때의 이벤트를 처리하는 함수

onTagNameInput()

사용자가 태그명을 입력했을 때 해당 태그를 가진 이미지의 목록을 ImageView에 보여주는 함수

onPeriodInput()

사용자가 촬영 날짜를 기간으로 입력했을 때 촬영 날짜가 해당 기간인 이미지의 목록을 ImageView에 보여주는 함수

onMapButtonClicked()

사용자가 구글 맵 버튼을 클릭했을 때 구글 맵 API로 사진 촬영 장소를 검색하는 기능을 제공하는 함수

onDiaryButtonClicked()

사용자가 이미지 검색 화면에서 Write a Diary 버튼을 눌렀을 때 일기 작성 화면으로 이동하는 함수

**일기 작성 화면**

onStoreButtonClicked()

사용자가 선택한 이미지에 대해 일기를 쓰고 store 버튼을 눌렀을 때 일기를 저장하는 함수

onModifyButtonClicked()

사용자가 일기 작성 화면에서 modify 버튼을 클릭했을 때 일기 수정 기능을 제공하는 함수

onDeleteButtonClicked()

사용자가 일기 작성 화면에서 delete 버튼을 클릭했을 때 일기를 삭제하는 함수

CheckBoxChecked()

체크박스에 체크가 되어 있지 않으면 일기 작성 화면을, 체크가 되어 있으면 일기 내용에서 키워드를 선택하여 태그를 추가로 생성 가능한 화면을 보여주는 함수

onCreateTagsClicked()

사용자가 create tags 버튼을 클릭했을 때 사용자가 선택한 키워드로 이미지 태그를 생성하는 함수

@@@ 함수 세부 설계 @@@

galleryInfoLink(){

갤러리 앱과 데이터베이스 연결

태그, 촬영 날짜 및 시간, 촬영 장소 정보(GPS 좌표) 연동하기

데이터베이스에서 GPS 좌표 정보를 주소 정보로 변환하기

}

onPrevButtonPressed(){

if(현재 이미지가 가장 첫 번째 이미지가 아니라면){

이전 이미지로 이동하기

}

Else {

화면에 “첫 번째 이미지입니다”라는 메시지를 Toast로 띄워주기

}

}

onNextButtonPressed(){

if(현재 이미지가 마지막 이미지가 아니라면){

다음 이미지로 이동

}

Else {

화면에 “마지막 이미지입니다”라는 메시지를 Toast로 띄워주기

}

}

onTagNameInput(){

사용자가 입력한 태그명을 가진 이미지를 데이터베이스에서 검색

검색한 이미지의 목록을 ImageView에 보여주기

}

onPeriodInput(){

if(시작 날짜 <= 끝 날짜) {

촬영 날짜가 사용자가 입력한 시작 날짜와 끝 날짜 사이인 이미지를 데이터베이스에서 검색

이미지의 목록을 ImageView에 보여주기

}

else {

“시작 날짜는 끝 날짜보다 이전이거나 끝 날짜와 같아야 합니다” 라는 메시지를 Toast로 보여주기

}

}

onMapButtonClicked(){

구글 맵 접속하기

사용자가 선택한 위치의 좌표 값을 주소로 변경하기

사용자가 선택한 위치의 주소와 촬영한 장소의 도/시/군(구) 정보가 모두 일치하는 이미지를 데이터베이스에서 검색

이미지의 목록을 ImageView에 보여주기

}

onDiaryButtonClicked(){

일기 작성 화면으로 전환하기

}

onStoreButtonClicked(){

사용자에게 “일기를 저장하겠습니까? (yes/no)”라는 텍스트 창을 보여주기

If(사용자가 yes를 누른 경우) {

사용자가 작성한 일기를 데이터베이스에 저장

}

Else{

일기를 저장하지 않은 채로 텍스트 창 닫기

}

}

onModifyButtonClicked(){

while(사용자가 store 버튼을 누르기 전까지){

사용자 일기 수정 기능 제공

}

}

onDeleteButtonClicked(){

사용자에게 “일기를 삭제하겠습니까? (yes/no)”라는 텍스트 창을 보여주기

If(사용자가 yes를 누른 경우) {

데이터베이스에서 일기 삭제

이미지 검색 화면으로 돌아가기

}

Else {

일기를 삭제하지 않은 채로 텍스트 창 닫기

}

}

CheckBoxChecked(){

If(체크박스에 체크가 되어 있지 않다면){

일기 작성 화면을 보여주기

0 반환

}

Else {

키워드 생성 화면을 보여주기

1 반환

}

}

onCreateTagsClicked(){

if(CheckBoxChecked()의 반환 값이 0이면){

함수 종료

}

Else {

if(사용자가 선택한 키워드가 없을 경우) {  
 “태그를 생성할 키워드를 1개 이상 선택해주세요”라는 메시지를 Toast로 보여주기

}

Else {

사용자가 선택한 키워드로 태그 생성 후 데이터베이스에 저장

}

}

}