|  |  |
| --- | --- |
| **실습 주제** | 데이터베이스를 이용한 맛집 |
| **이름** | 이충헌 |
| **학번** | 1594029 |
| **학과** | IT응용시스템공학과 |

# 자기평가

|  |  |
| --- | --- |
| **과제 항목 ID** | **평가** |
| 1 | 맛집 목록 액티비티 구현 완성 |
| 2 | 맛집 등록 액티비티 구현 완성 |
| 3 | 맛집 상세정보 액티비티 추가구현 완성 |
| 4 | 메뉴 등록 액티비티 구현 완성 |
| 5 |  |

# 결과

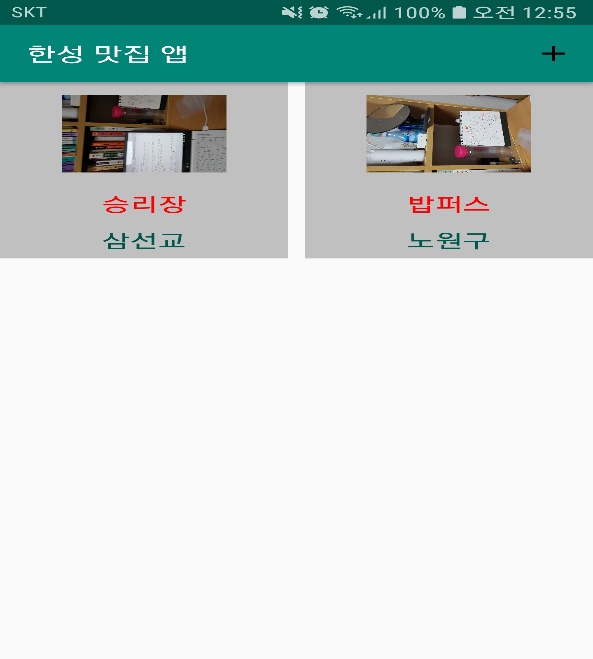
|  |  |
| --- | --- |
| 과제 항목 ID | *(예시, 1)* |
| 과제 내용 | *(예시, UI 레이아웃 구성)* |

결과 설명 및 결과 화면 캡처

0. 구현 방법

- 언어는 java를 이용했고, 데이터베이스는 SQLite(내부 데이터베이스)를 이용했습니다.

1. 맛집 목록 액티비티 구현



- 어플을 시작했을 때 처음보이는 RestaurantCatalog 엑티비티입니다. 뷰 크기를 일정하게 넣을 수 있고 옵션 앱 바로 RestaurantRegister 액티비티로 이동할 수 있습니다.

이 액티비티를 사용하기 위해 DBHelper를 사용했습니다. 맛집 데이터베이스에선 식당의 이름, 위치, 번호, 영업시간, 이미지를 지정할 수 있게 정의하였습니다. 커서로 DB레코드를 접근하기 위해 Cursor를 만들어서 하나씩 접근할 수 있습니다.

RestaurantDbHelper = new DBHelper(this); //맛집데이터베이스를 이 액티비티에서 사용하기 위해 객체생성합니다.

- SimpleCursorAdapter를 이용해서 객체하나하나를 xml요소에 맞게 간단하게 저장할 수 있습니다. 그리고 맛집을 클릭했을 때 RestaurantDetail 액티비티로 이동할 수 있도록 setOnItemClickListener를 생성하고 인텐트는 식당의 ID요소만 넣고 putExtra해줍니다. 왜나하면 맛집ID 하나로 식당을 식별이 가능해서 받는 쪽에서 커서를 이용하면 모든 속성들을 불러올 수 있기 때문입니다.

- 이 문제를 통해서 그동안 리스트 뷰 만을 써왔었는데 그리드 뷰를 사용했기 때문에 안드로이드의 그리드 뷰에 대한 지식이 늘게 되었고 그 전의 과제1은 미리 정보를 액티비티에 저장해서 과제를 수행했지만 이 과제2는 데이터베이스를 응용하여 데이터베이스를 생성하면서 만들어야 하기 때문에 어려웠습니다.

- 그리고 문제 1,2,3,4번이 모두 데이터베이스가 관련된 문제들이었기 때문에 데이터베이스를 소홀히 할 수 없었습니다. 그래서 차근차근 동영상 강의와 튜터링을 통해 데이터베이스를 공부했고 어느 정도 할 수 있게 되었습니다. 이 문제는 정보를 하나씩 넘겨주지 않고 ID만을 이용해서 넘길 수 있다는 것이 신기하고 재미있었습니다. 그리고 스마트폰에서 편리한 역할을 하는 옵션 앱 바의 사용으로 편리하게 액티비티 이동이 가능한 것이 좋았고 매니페스트를 제대로 선언해주지 않으면 오류가 나온다는 것을 알게 되었습니다.

2. 맛집 등록 액티비티 구현



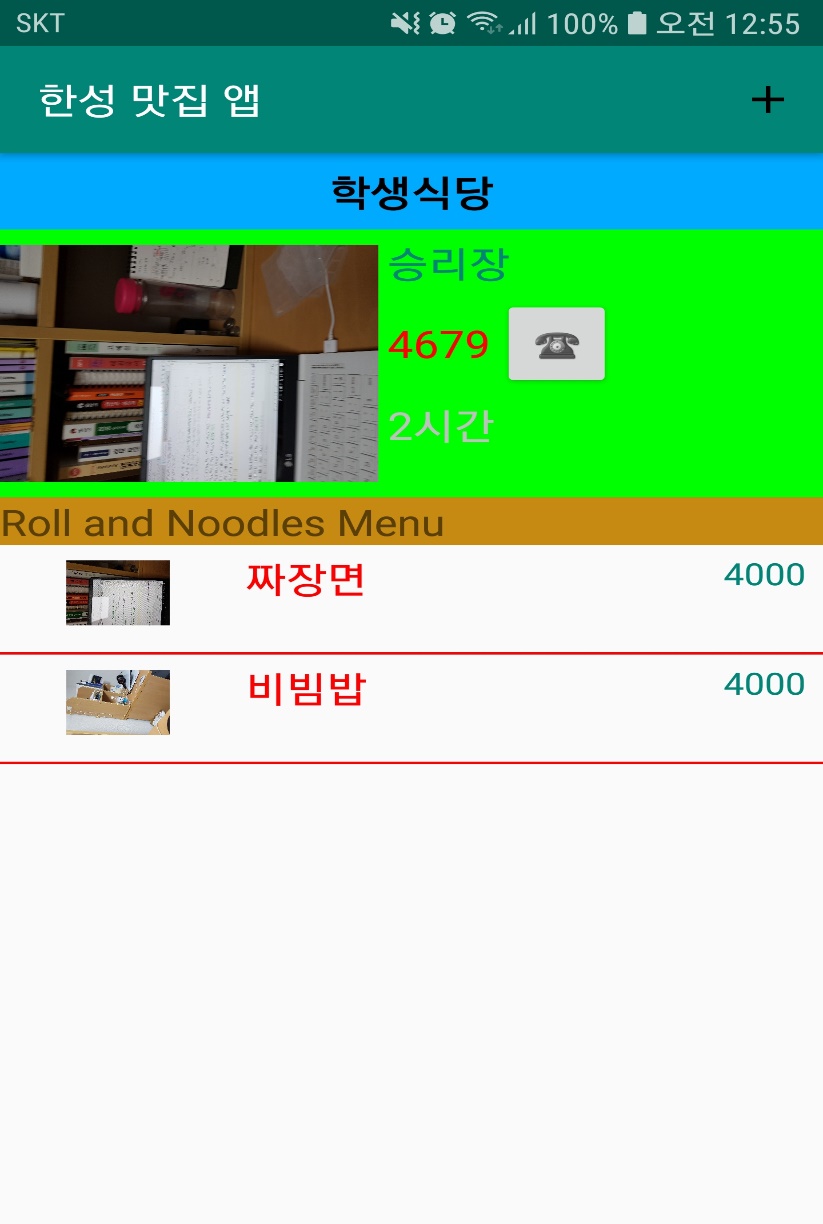
- 맛집 등록 액티비티 역시 맛집 데이터베이스를 이용하여 데이터베이스 저장을 위주로 작업하였고 checkDangerousPermissions()를 이용해서 사용자에게 권한을 물어보게 하였고 보안기능이 좋아졌습니다. 맛집 등록 버튼을 누르면 그 전의 액티비티인 RestaurantCatalog로 돌아가게 해놨습니다. 사진찍기 버튼을 만들었는데 이미지Uri를 이용하여 이미지를 Uri로 만든 다음에 액티비티 간 이동이 용이하게 만들었습니다.

- 그리고 삽입버튼을 누르면 작성한 정보들이 데이터베이스에 저장이 되게 insertUserByMethod()를 이용하여 데이터베이스 삽입 구현을 하였습니다.

찍은 사진은 사진을 찍고 확인을 누르면 이미지버튼에 나오게 구현했고 그 이미지 역시 Uri로 변환해서 작업하였습니다.

- 두번째 문제의 가장 어려웠던 점은 카메라로 찍은 것을 어떻게 다시 이미지버튼에 삽입이 되는지가 가장 궁금한 내용이었습니다. 액티비티 간에 이동은 인텐트로 편리하게 주고받을 수 있는데 카메라사진은 문자나 숫자처럼 있는 그대로 주고받지 않고 URI로 변환하면서 주고받아야 하는 것이 이 과제의 어려웠던 점이었습니다. 그러나 전에 배웠던 onActivityResult() 메소드와 사진이 저장되었는 sdcard경로만 제대로 공부했다면 쉽게 해결할 수 있는 문제들이 었고 이미지 저장도 매니페스트가 중요했다. 매니페스트에 선언을 안해서 오류가 나고 시간을 소모했던 적이 많이 있었고 안드로이드는 기본이 가장 중요하다고 느꼈습니다.

3. 맛집상세정보 액티비티 추가 구현



- 데이터베이스는 맛집 데이터베이스와 메뉴 데이터베이스 2개가 있어야 완성 가능합니다. resId는 식당을 식별하기위한 ID변수입니다. 이 변수를 활용해서 맛집의 이름, 번호, 영업시간, 위치를 모두 얻어올 수 있고, ID에 딸린 메뉴를 삽입할 때도 중요한 역할을 합니다.

- cursor.moveToFirst() 그리고 커서의 위치를 맨 앞으로 오게 하기 위한 메소드가 없으면 데이터베이스가 위치를 식별하는데 어려움이 많습니다.

- name.setText(cursor.getString(1)) 커서를 사용하여 column에 따라서 쉽게 데이터베이스 접근이 가능합니다.

- img.setImageURI(Uri.parse(cursor.getString(5)))로 통해 이미지는 다른 정보와는 약간 다르게 set한다는 것을 알 수 있습니다.

- viewAllToListView\_MENU() 입력된 메뉴를 불러오는 메소드입니다.

cursor2 = MenuDBHelper.getAllUsersBySQL(resId)는 커서2가 사용되고 메뉴 데이터베이스를 이용합니다. 이미지와 이름, 가격을 불러와서 화면에 보여지게 합니다. RestaurantCatalog처럼 옵션앱바를 사용하여 누르면 MenuRegister액티비티로 이동하게 하고 안에 finish()를 넣어서 좀더 액티비티간 입력사이에 깔끔하게 이동할 수 있도록 했습니다.

- 이 문제는 가장 나중에 완료한 문제입니다. 왜냐하면 데이터베이스가 2가지 사용되어야 하고 잘못했다간 메뉴와 맛집이 서로를 인식 못할 수도 있기 때문에 오류가 많이 나오고 시간이 많이 걸렸습니다. 맛집ID를 알맞게 불러오고 메뉴데이터베이스에 맛집의 ID를 정의해줘서 데이터베이스 테이블이 유기적으로 연결될 수 있도록 설계해주는 것이 중요합니다.

- 그리고 액티비티가 많아짐에 따라 변수의 이름을 선언할 때도 식별이 잘될 수 있도록 하는 것이 나중에 파일을 보거나 발표를 할 때 헷갈리지 않은 것 같습니다. 그리고 Cursor.moveToFirst()를 지정하지 않아서 오류가 많이 났습니다. 이 메소드가 없으면 데이터베이스의 커서가 길을 잃는 현상이 발생해서 정확한 정보를 찾지 못합니다. 그래서 꼭 선언하여야 커서가 처음위치부터 이동할 수 있다.

4. 메뉴등록 액티비티 구현



- 맛집등록과 마찬가지로 카메라를 이용합니다. 특이한 점은 맛집ID가 있어야 맛집에 맞는 메뉴 삽입이 가능합니다. 만약 맛집ID가 없으면 어떤 맛집을 선택해도 같은 메뉴가 보여지게 됩니다.

- 역시checkDangerousPermissions()메소드로 보안성을 좋게 하고

Intent intent4 = getIntent();

resto = intent4.getIntExtra("dataFromRestaurantDetail",-1) RestaurantDetail 액티비티에서 식당의 id값을 불러와줍니다.

- 삽입버튼을 누르면 RestaurantDetail액티비티로 이동하게 됩니다.

이미지 삽입버튼도 위에 맛집 삽입과 같이 해주면 된다. 이때 변수명이 맛집과 헷갈리지 않게 조심합니다

- onActivityResult()는 찍은 사진을 이미지버튼에 저장되게 합니다.

- 이미지를 삽입할 때는 경로가 틀리지 않는지 확인하면서 해야합니다.

- 마지막문제는 맛집등록과 거의 똑같기 때문에 맛집등록을 잘 해냈다면 쉽게 할 수 있는 문제였습니다. 그러나 가장 중요했던 것은 맛집ID를 이용하여 알맞게 메뉴가 저장될 수 있는 것이 중요했습니다. 이것을 숙지하지 않는다면 오류가 많이 나오게 됩니다. 메뉴데이터베이스를 주로 사용하면서 변수의 이름을 헷갈리지 않는 것이 중요한 문제였다.

5. 총평

- 이 과제를 끝내면서 안드로이드에서 가장 중요하다고 생각된 것은 데이터베이스라고 생각합니다. 왜냐하면 그때그때 저장하거나 삭제하거나 수정하는 것에 모두 관여하게 때문에 데이터베이스가 가장 중요하고 ID하나만으로도 저장된 정보를 불러오는 것이 코드의 효율성을 봐도 좋은 방법을 배웠다고 생각합니다. 그 전까지는 모든 정보를 하나하나 인텐트 해가면서 넣어줬었는데 이제는 편리하게 값을 받아올 수 있습니다.

- 그리고 오류가 나올 때는 매니페스트를 먼저 확인해 주는 것도 필요합니다. 왜냐하면 모든 것이 완벽하다고 생각했는데 오류가 나서 매니페스트를 확인 해봤는데 선언 한 줄을 안 해줘서 시간이 걸린 적이 많이 있었습니다. 그리고 과제1과 연계가 되는 것이 좋았고 카메라의 사용은 경로설정을 잘해주고 String요소와는 다르게 Uri로 이동하는 것이 신기했습니다. 튜터에게 물어보니 이미지 자체의 이동은 불가해서 Uri로 변환해준다고 했습니다. 그리고 옵션 앱 바를 통해 편리하게 앱 바에서 이동하는 것이 실무에서 자주 쓰일 것 같았고 이 과목을 통해 안드로이드의 재미를 알게 해주었고 객체 지향 프로그래밍이 이렇게도 쓰일 수 있구나 하고 생각이 들었고 한 학기가 유익하게 흘러갔습니다.