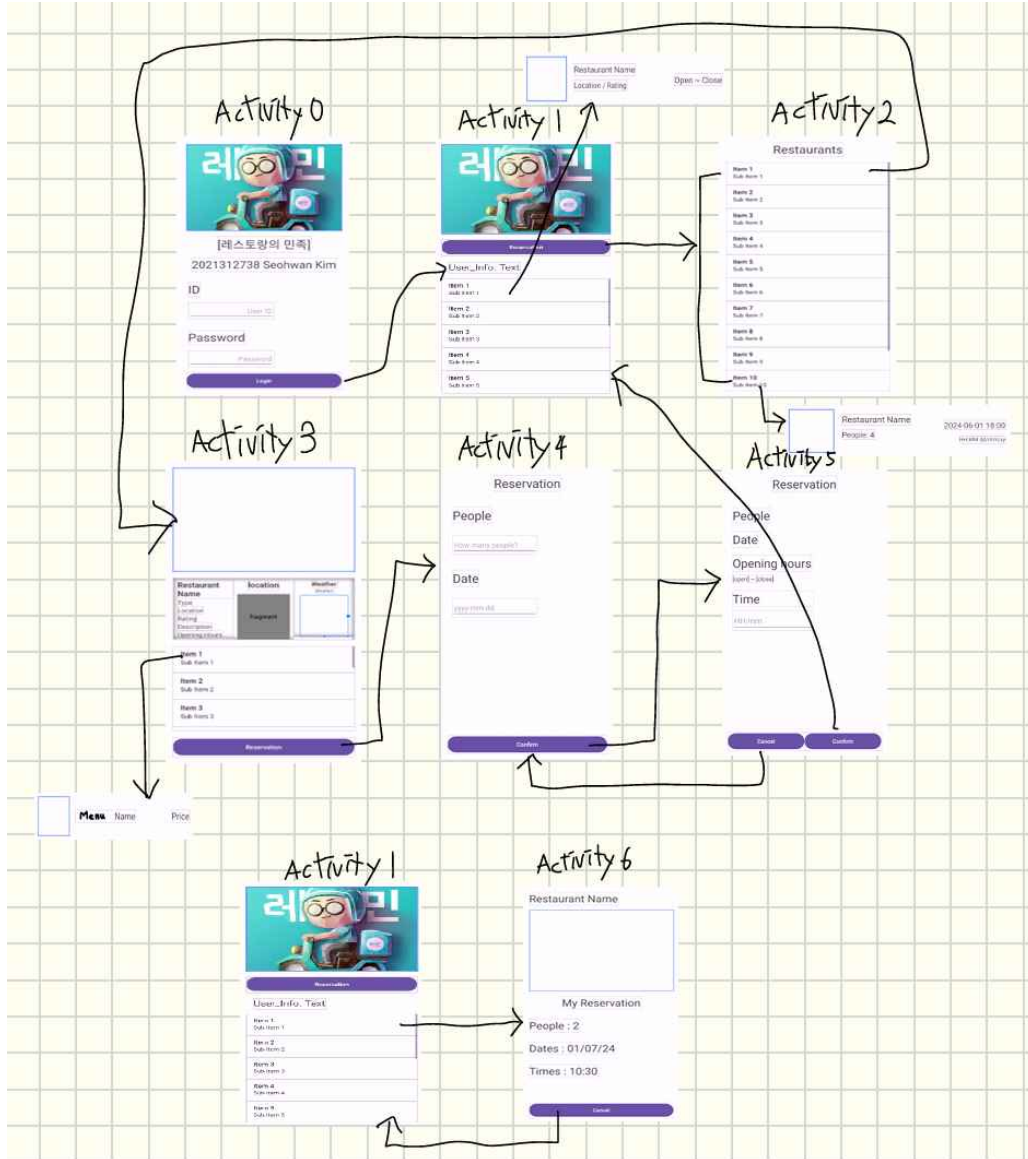


Mobile App Programming Final Project Report

2021312738 소프트웨어학과 김서환

1. Spec1 (Low-fidelity prototype)



2. Spec2 (Detailed descriptions for each activity)

- User.kt, Restaurant.kt, JSONUtils

우선 restaurant_info.json, user_info.json 파일을 assets폴더에 저장하고 User.kt, Restaurant.kt 파일에 각 데이터를 관리하기 위한 클래스를 생성하여 유저와 레스토랑에 대한 정보를 읽고 쓰는 과정을 용이하게 했다. 또한, JsonUtils라는 object파일을 만들어 JsonUtils.loadUsers, JsonUtils.loadRestaurants, JsonUtils.saveUsers의 함수를 통해 user_info.json에서 데이터를 읽거나 저장하고, restaurant.json에서 데이터를 읽었다.

레스토랑은 수정될 이유가 없기에 write에 대한 함수는 구현하지 않았다.

이 과정에서 loadUsers, saveUsers 함수를 구현하는 과정에서 assets 폴더를 직접 접근하여

수정하고 업데이트된 내용을 반영하려고 하니까 반영이 계속 안되는 현상이 나타나서 찾아보니 assets 폴더 하위의 있는 파일들은 오직 read only라는 것을 알 수 있었다.

=> 그래서 내부 저장소를 이용하는 방법을 선택했다.

loadUsers에서 처음에 user_info.json 파일이 없다면 내부 저장소에 user_info.json 파일을 만들어 assets 하위에 있는 user_info.json를 copy해서 내부 저장소로 가져온다. 만약 user_info.json가 내부 저장소에 존재한다면 이 과정을 생략한다.

- xml files

xml 파일은 activity 0(main activity), activity 1 ~ 6, reservation_list에 들어갈 item, restaurant_list에 들어갈 item, menu_list에 들어갈 item 이렇게 총 10개의 xml 파일을 pdf에 존재하는 내용과 비슷하게 구현했다.

- MainActivity(Activity 0)

Activity 0에서는 앱의 제목에 맞는 이미지, 앱의 제목, 본인 학번과 이름, 로그인Id 입력란, 비밀번호 입력란, 로그인 버튼으로 구성했다. 로그인 버튼을 누르면 user_info.json에서 입력 받은 userid와 password를 비교하여 두 조건을 동시에 만족한다면 Intent를 통해 activity 1을 실행하도록 구현했고, 실패 시에 toast 메시지를 출력하도록 구현했다.

- Activity 1

이전 activity에서 로그인에 성공하면 activity 1을 실행하고, activity 1에서는 intent로 전송된 userid를 user_info.json에서 비교하여 같은 id를 지닌 user를 currentUser로 설정해줬다. 해당 유저의 예약 목록을 ListView로 표현해야 하기 때문에 adapter를 이용해서 각각의 reservation이 ListView에 담겨서 표현될 수 있도록 구현했다. 또한, 각각의 reservation은 클릭시에 해당 reservation에 대한 정보와 해당 user에 대한 정보를 intent로 넘겨서 activity 6을 실행하도록 구현했다. 마지막으로 Reservation 버튼을 누르게 되면 intent로 해당 user의 id를 넘겨서 activity 2를 실행하도록 구현했다. USER_ID를 계속 넘기는 이유는 나중에 activity 5에서 user_info.json에서 해당 user의 데이터에 reservation을 새로 추가해줘야 하기 때문에 계속 넘기는 것이다. 또한 activity 1에서는 back 버튼을 누르면 앱이 종료되게끔 구현했다.

- Activity 2

activity 2에서는 예약하고 싶은 레스토랑을 선택할 수 있는 화면인데, 이전 activity에서 예약 목록에 대한 ListView를 구현하는 것과 마찬가지로, Restaurant에 대한 목록을 adapter를 이용하여 ListView로 구현했다. 각각의 restaurant은 레스토랑 이미지, 이름, 위치, 평점, 영업시간을 보여준다. 또한, 각각의 restaurant를 클릭시에 해당 restaurant에 대한 정보와 해당 user에 대한 정보를 intent로 넘겨서 activity 3을 실행하도록 구현했다. pdf에 나와있듯이 activity 1에서는 back 버튼을 누르면 앱이 종료되게끔 구현했지만, activity 2~6까지는 back 버튼을 누르면 이전 activity로 돌아가도록 구현했다.

- Activity 3

선택된 레스토랑의 id를 restaurant.json에서 비교하여 값이 같은 레스토랑을 찾고, 이전 activity들과 마찬가지로 해당 레스토랑의 menu들에 대한 ListView를 구현했다. menu List의 각각의 menu들은 메뉴 이미지, 이름, 가격을 보여준다. Reservation 버튼을 누르면 Activity 4가 실행된다. (추가기능을 제외한 설명)

- Activity 4

activity 4는 예약에 대한 자세한 사항을 결정하는 화면인데, 같이 올 사람의 수와 날짜를 사용자로부터 입력받는다. 사람의 수가 1~10일때만 valid하고, 나머지는 invalid하다. valid한지 안한지 체크하는 문법을 조건문을 통해 구현했고, 날짜 또한 오늘 이후의 날짜인 경우에만 valid하고, 나머지는 invalid하다. 날짜 또한 valid한지 안한지 체크하는 문법을 조건문을 통해 구현했다. invalid한 값을 입력하면 toast 메시지를 출력하고 activity가 바뀌지 않고 계속 입력값을 수정하거나 버튼을 누를 수 있도록 `return@setOnClickListener`를 이용해서 만들었다. 추가적으로 중요한 것은 아니지만, 사용자가 어떤 형식의 시간 형식을 입력해야하는지 알 수 있도록 EditText의 hint로 형식에 대한 힌트를 알려줬다. 그리고 사람의 수, 날짜 모두 valid한 값으로 잘 입력 받았을 경우에만 Confirm 버튼을 누르면 RESTAURANT_ID, NUMBER_OF_PEOPLE, DATE, USER_ID를 intent로 전송하여 Activity 5가 실행되도록 구현했다.

- Activity 5

이전 activity 4가 보낸 intent에 있는 내용들을 잘 받아서 화면에 출력하고, activity 5에서는 몇 시에 식사하는 것으로 예약을 확정할지에 대한 것을 결정하는 단계이다. Time은 해당 레스토랑의 영업 시간 사이의 값이 되어야하며, 이 또한 invalid한 값에 대해서는 activity가 바뀌지 않고 계속 입력값을 수정하거나 버튼을 누를 수 있도록 `return@setOnClickListener`를 이용해서 만들었다. 만약 valid한 값을 유저가 입력하고 confirm 버튼을 누르게 되면 내부 저장소에 있는 user_info.json에 있던 해당 유저의 데이터에 덮어씌우고 예약이 확정되었으므로 intent 필터의 flag를 설정하여 기존에 실행중이던 모든 activity들을 종료시키고, 새로운 task 스택에 activity 1이 실행되도록 구현했다. intent를 통해서 Activity 1로 돌아가게 된다. 반면, cancel 버튼을 누르게 되면 Activity 4로 다시 돌아가도록 구현했다.

- Activity 6

activity 6은 activity 1에서 특정 reservation을 클릭했을 때 출력되는 화면으로 activity 1로부터 intent로 전달 받은 데이터를 참고하여 레스토랑의 이름, 썸네일, 예약된 같이 오는 사람의 수, 날짜, 시간 등을 화면에 출력한다. 그리고 Cancel 버튼을 누르면 예약을 취소하게 되고, 예약이 취소되었음을 알리는 toast 메시지를 출력한다. 또한, 다시 Activity 1으로 돌아갔을 때 취소한 예약을 ListView에서 삭제해야 한다.

전체적으로 Activity 사이의 관계가 명확하기에 명시적 Intent를 사용하여 구현했고, ListView에 반복되는 요소들이 들어가기때문에 adapter를 사용하여 구현했다.

3. Additional Functionality 추가 기능

추가 기능을 3가지 추가했다.

첫 번째 추가 기능은 Activity 3의 중간 부분의 레스토랑에 대한 디테일 정보를 출력하는 부분에 Google Map API와 연동하여 사용자에게 레스토랑 위치에 대한 구글 맵 지도를 fragment로 붙여서 직관성을 더 강조했다.

(하지만 이 기능에 조교님께 양해드리고 싶은 점이 있습니다. 기능이 잘 작동하는데 Google Map API를 네트워크를 통해 연동하여 가져오는 부분에 있어서 앱을 처음 실행했을 때 네트워크 이슈로 잘 못가져오는 현상이 가끔 생깁니다.

그래서 처음 앱을 실행했을 때 Google Map API가 데이터를 가져오다가 네트워크에 의해 문제가 생겨 initial address에서 멈춰있다면 back버튼을 누르셨다가 다시 들어오시면 잘 작동

함을 알 수 있으실 겁니다 !

-> 네트워크 이슈라서 delay나 success flag를 통해서 처리해봤는데 어떨때 처음 앱을 실행해도 잘 가져오지만, 아주 가끔 못 가져오는 경우가 있어서 이 점 양해 부탁드립니다.)

두 번째 추가 기능은 Activity 3에서 위의 Google Map을 통해 구현한 fragment 옆에 해당 레스토랑 위치에서의 날씨에 대한 설명과 날씨에 따라 쉽게 연상되는 이미지를 같이 배치하여 사용자에게 풍부한 정보를 줄 수 있도록 구현했다.

세 번째 추가 기능은 앞서 설명했지만, assets 폴더에서 작업하는 것이 아닌 안드로이드 앱의 내부 저장소에 데이터를 저장하여 읽고 쓰는 작업을 진행할 수 있도록 했다.

그렇다면 왜 Google Map API와 Open Weather API, 내부 저장소를 추가 기능으로 선택했는지 이유를 설명해보면,

첫 번째로는 처음에 Activity 3을 구현했던 것은 단순히 레스토랑 이름, 위치, 평점, 설명 등의 모두 텍스트로 이루어진 정보로만 Activity 3이 구성되었었다. 하지만, Google Map API를 이용하여 사용자가 더 직관적으로 레스토랑의 대략적인 위치를 파악할 수 있도록 하고, API를 통해 만들어진 지도의 fragment에서 해당 레스토랑의 위치를 클릭하면 Google Map이 바로 연동되어 커지는 것을 통해서 가고 싶은 레스토랑 주변의 볼거리나 놀거리 등을 찾아볼 수 있는 부가기능을 더하고 싶어서 구현했다.

두 번째로는 Open Weather API를 이용하여 사용자가 해당 레스토랑의 위치에서의 날씨를 유저가 파악할 수 있도록 하여 날씨에 더 적합한 메뉴를 고를 수 있도록 하거나 날씨에 맞는 옷을 입고 갈 수 있도록 하기 위해서 구현했다.

세 번째로는 내부 저장소를 활용함으로써 데이터를 쉽게 저장하고 쉽게 읽을 수 있으며 앱을 종료하더라도 내부 저장소에 데이터가 유지가 되기 때문에 앱을 다시 실행해도 데이터가 그대로 존재하도록 구현하여 사용자 경험, 지속성, 신뢰성을 향상시키기 위해 내부 저장소를 활용해서 구현했다.

그렇다면 각 기능을 어떻게 구현했는지 설명해보면,

첫 번째 기능은 onMapReady함수(구글 맵이 준비되면 호출되는 콜백 메서드) 안에 레스토랑의 location 값으로 marker를 추가하고, 해당 location으로 카메라 무빙을 옮기는 함수를 호출함으로써 출력되는 Google Map이 레스토랑의 위치를 가리키도록 구현했다.

두 번째 기능은 직접 openweatherapi에 요청하는 url을 작성하여 받은 응답 메시지를 Json 구조로 변환하여 날씨에 대한 설명, 온도 등의 데이터를 받아와 온도 값에 따라 ImageView를 달리 설정해줌으로써 구현했다.

세 번째 기능은 loadUsers함수 안에서 File(context.filesDir, "user_info.json")을 사용했다. loadUsers(context) => 이렇게 호출해주면 알아서 내부 저장소에 있는 user_info.json을 참조하여 작업을 진행한다. 만약 파일이 없다면 assets 폴더에 있는 user_info.json을 복사하여 내부 저장소로 저장하고 내부 저장소에 저장된 user_info.json을 이용하여 작업을 진행하도록 구현했다.

(Demo Video는 Android screen recording으로는 제 목소리가 녹음되지 않아서 앱의 동작 과정을 제가 설명하는데 있어서 말로 설명하는 부분이 추가되는 것이 더 나을 것 같다고 판단해서 캡처 도구를 이용해서 녹화했습니다!)