

Project Report

모바일 소프트웨어 프로그래밍

프로젝트 내용 : pdf뷰어와 사전을 결합한 앱 개발

목차

- 1 문제 개요
- 2 요구 사항
- 3 설계 과정
- 4 구현 과정
- 5 프로젝트 실패 원인 자체 분석
- 6 자체 보완 계획
- 7 결론, 소감

16조

201002511

최정현

1. 문제 개요

pdf뷰어를 볼 때 영어 단어의 모르는 뜻이나 뜻을 정확히 알고자 하기 위해서는 pdf뷰어에서 홈으로 돌아와 사전을 킨 다음 단어를 복사, 붙여넣기를 해야 한다. 만약 단어를 까먹기라도 한다면 다시 pdf뷰어를 작업 표시줄에서 고르는 모습을 볼 수 있었다. 이러한 번거로움을 방지하기 위해 앱을 만든다.

2. 요구 사항

pdf뷰어 기능을 위해서는 먼저 사용자 SD카드 내의 폴더와 pdf파일만을 읽어 들여 사용자가 선택하기 편리하게 해 주어야 한다. 사용자는 보통 pdf파일을 폴더를 이용해 읽고 관리한다.

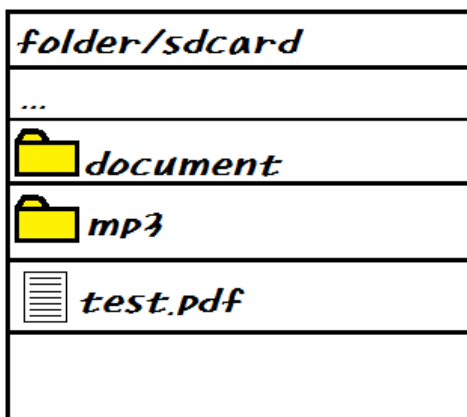
사전의 경우에는 초기 상태에서는 외부 인터넷을 사용하여 검색하는 방식을 사용하며, 사용자가 원할 경우 사전을 다운받아 오프라인 상태에서도 사용할 수 있도록 한다.

기본적인 요구사항을 만족시킨 다음 그 외의 요구사항은 추가적으로 파악하고 추가할 예정이다.

3. 설계 과정

- UI설계

a. pdf파일 선택 UI



사용자가 앱을 시작했을 때의 UI이다. “...”을 통해 이전 폴더로 이동할 수 있으며 폴더를 눌러 해당 폴더로 이동할 수 있다. pdf파일을 눌렀을 시에는 pdf파일이 열리게 된다.

b. pdf뷰어 UI

It is rather for us to be here dedicated to the great task remaining before us—that from these honored dead we take increased devotion to that cause for which they gave the last full measure of devotion—that we here highly resolve that these dead shall not have died in vain—that this nation, under God, shall have a new birth of freedom—and that government of the people, by the people, for the people, shall not perish from the earth

위의 내용을 pdf파일이라고 가정하면 위와 같이 화면에 보여주게 된다. 여기서 단어를 선택하게 되면 아래와 같다.

c. 단어 선택시 UI

It is rather for us to be here dedicated to the great task remaining before us—that from these honored dead we take increased devotion to that cause for which they gave the last full measure of devotion. **복사** we here highly resolve that these dead shall not have died in vain—that this nation, under God, shall have a new birth of freedom—and that government of the people, by the people, for the people, shall not perish from the earth

단어를 길게 터치하면 나오는 화면이다 복사를 누르게 되면 해당 문자를 클립보드에 복사하고, 단어찾기를 누르면 아래와 같이 프래그먼트가 나오게 된다.

d. 단어찾기 선택시 UI

It is rather for us to be here dedicated to the great task remaining before us—that from these honored dead we take increased devotion to that cause for which they gave the last full measure of devotion—that we here highly resolve that these dead shall not have died in

devotion

- ① 헌신, 전심, 전념
- ② 귀의, 신앙심

단어찾기를 누르게 되면 화면 하단의 프래그먼트가 나타나게 된다. 이 프래그먼트는 프래그먼트 이외의 화면 아무 곳을 터치하면 사라지게 된다.

4. 구현 과정

- pdf파일 리스트는 어떻게 보여줄 것인가?

사용자의 SD카드에 있는 폴더 리스트와 함께 pdf파일만을 보여준다. 이는 ListView와 manifest 파일에서 SD카드 제어 권한을 허용하게 한다. 파일 목록을 필터링 하게 되면 SD카드에 있는 폴더와 pdf파일을 모두 찾아낼 수 있다. 그 다음 폴더와 pdf파일을 눌렀을 때를 구분하여 서로 다른 액션을 만든다.

```
//리스트 아이템을 클릭했을시, 폴더일 때와 pdf파일일 때를 다르다.
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
    String strItem = mArFile.get(position);
    if(strItem.endsWith(".pdf")){
        //pdf파일을 눌렀을 시
        //Intent에 보낼 파일의 절대 경로를 가져온다.
        strItem = getAbsolutePath(strItem);
        //pdfViewActivity에 보낼 Intent를 생성한다.
        Intent intent = new Intent(MainActivity.this, pdfViewActivity.class);
        //pdfFile변수에 strItem을 저장하고
        intent.putExtra("pdfFile", strItem);
        //플래그를 추가한다.
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK|Intent.FLAG_ACTIVITY_CLEAR_TOP);
        //인텐트를 이용해 새 액티비티를 생성한다.
        startActivity(intent);
    }else{
        //파일일 경우 폴더 파일의 리스트를 보여주게 된다.
        Log.i("Dic", "I'm directory"+strItem);
        String strPath = getAbsolutePath(strItem);
        //파일 리스트를 가져오게 된다.
        String[] fileList = getFileList(strPath);
        //리스트 출력
        fileList2Array(fileList);
    }
}
```

- pdf파일은 어떻게 보여줄 것인가?

Rolipop에서는 pdfRenderer클래스를 따로 제공해주나, 현재 기기는 KitKat버전이었기 때문에 실험이 불가능 했다. 따라서 Kitkat에서 동작하는 방법에 대해 생각을 해 보았다. 방법은 아래와 같다.

1. Java를 이용해 pdf를 렌더링 하는 방법

Java에서 사용하는 pdf 렌더링 라이브러리를 활용한다. 라이브러리를 다운받고 사용하게 되면 pdf를 byte 코드로 바꾼 후 이미지화 시켜서 화면에 출력하는 방식이다.

2. 외부 라이브러리를 사용하는 방법

기준에 만들어진 라이브러리를 사용한다. 일부 라이브러리는 pdf를 이미지로 렌더링 하는 경우도 있는가 하면, 심지어는 확대 축소도 별도의 버튼을 눌러야만 되는 경우도 있었다. 다른 라이브러리를 알아보려고 하였지만 여러 가지 제약조건과 번거로운 절차가 존재하고 심지어는 유료 라이브러리까지 있었다.

3. 사용한 라이브러리에 대한 설명

이번에 선택한 방식은 Java를 이용해 pdf를 렌더링 하는 방법을 선택하였다. pdf파일을 받게 되면 base64 인코딩을 하여 이미지화 시키는 방법이다. 문제점은 화면이 깨지고 문자열 선택이 불가능 하였다.

- pdf파일을 텍스트를 선택할 수 있게 보여줬다면, 사전은 어떻게 보여줄려고 하였는가?

텍스트를 클립보드에 복사한 상태에서 인터넷을 통한 사전검색이었다면 복사한 텍스트를 넣고 검색하여 프래그먼트를 띄우는 방식을 선택하려고 하였다.

5. 프로젝트 실패 원인 자체 분석

1. 스마트 기기 사용 경험이 적다.

스마트기기를 사용하게 된 것은 얼마 되지 않았다. 그렇기 때문에 눈앞에 있는 문제만 볼 수밖에 없었다. 그것이 바로 자주 쓰는 앱중 하나인 pdf뷰어였고, pdf뷰어에 관해서 밖에 생각뿐이 없었던 것이 첫 번째 원인이다.

2. 주제에 대해 실현 가능성에 대해 생각하지 못했다.

보통 주제를 생각할 때 현재 인원수와 구현 기간, 구현 환경에 대해 고려를 해야만 했다. pdf뷰어의 경우 현재 사용기기의 상위 버전인 Lolipop에서나 되며, 그 이하 버전에서는 pdf파일을 생성해주는 클래스만이 존재하였다. 비록 Adobe사의 pdf리더도 있지만, 복잡한 라이브러리와 심지어는 유료 라이브러리를 사용하는 리스크가 있었다.

3. 문제 해결에 사용할 라이브러리를 찾지 못하였다.

대부분의 OpenSource들을 사용하는 방법이 미숙하였으며, OpenSource를 받아 사용한다 하더라도 pdf파일을 이미지로 랜더링 하여 텍스트를 선택할 수 없는 상황이 발생하며, 심지어는 화면 확대를 위해서 버튼을 누르거나 화면에 워터마크를 만드는 경우도 있었다. 결국 원하는 라이브러리를 찾지 못 한것이 주된 원인이라 생각한다.

6. 자체 보완 계획

1. 스마트 기기의 기능에 대해 좀 더 알아본다.

스마트기기를 좀 더 가까이 한다. 스마트기기도 누가 사용하느냐에 따라 그냥 일반 전화기가 되거나 강력한 도구가 된다.

2. 주제의 실현 가능성에 대해 결정하기 전에 한 번 더 생각한다.

주제가 결정되면 한 번 더 검토를 하고 결정하여 무리하게 진행하거나, 불가능한 일을 진행하는 일이 없도록 한다.

7. 결론, 소감

안드로이드를 활용한 첫 프로젝트였지만 실패로 끝난 것은 매우 아쉽다. 하지만 실패했다고 해서 끝내는 것은 아니다. 한 번 실패했다고 이 프로젝트를 포기하는 것은 한 학기동안 배운 것을 날리는 것과 다름없다고 본다. 비록 프로젝트는 끝났으나 다시 진행하여 마무리를 지을 생각이다.