

Dynamic Digital Picture Frame

지도교수 : ○ ○ ○ 교수님

지도조교 : ○ ○ ○ 조교님

Team : ○ ○ ○

○ ○ ○ ○ ○ ○ ○ ○ ○ ○

1. 연구 목표

소형기기를 이용한 효율적인 DPF (Digital Picture Frame) 구현

DPF는 디지털 사진을 출력할 필요 없이 액자처럼 볼 수 있도록 해 주는 장치로서, 다양한 목적으로 사용된다. 예를 들어, 책상 위나 벽 등에 두어 장식용으로 사용하거나, 공공 장소에 설치하여 광고용으로 사용된다.

2. 연구의 필요성

- o. 현재 소형기기인 PandaBoard내에서 만들어 사용하고 있는 DPF(Digital Picture Frame)은 서버 소형기기간 미러링 방식으로 이루어 지고 있다. 이점은, 여러대의 소형기기를 사용 할 때에도 동일한 이미지만 전송하고 출력하는 문제를 가지고 있다. 이런 방식은 다양한 목적으로 사용하기가 매우 힘들다.

첫째, 각 클라이언트 마다 다른 이미지를 출력하기 위해서 여러대의 서버를 필요로 한다.
둘째, 사용자가 사진을 찍어 전송 할때에도 어떤 설정도 없이 전송되므로, 서버나 소형 기기에서 주제에 맞게 분류 하기가 매우 힘들다.

- o. 또한 현재 사용하고 있는 서버에서는 유지 관리기능이 없기 때문에, 프로그래머나 개발자가 아닌 사람이 관리하기가 매우 힘들고, 또한 개발자가 관리하기에도 매우 번거로운 상황이다.

3. 연구 내용

이번 프로젝트로 향상 시키고 싶은 내용은 서버 집중형 시스템이다. 데이터의 관리가 제대로 되지 않던 기존의 문제점을 분석하였고, 이를 바탕으로 데이터들의 관리와 분류가 이루어지도록 한다. 이번 프로젝트에서 크게 지향되는 사항은 아래와 같다.

o. 데이터 분류

데이터들의 테마, 태그별 분류를 바탕으로 데이터를 관리 할 수 있도록 한다. 이는 사용자에게 편리한 데이터 사용과 후에 DPF 연결에 있어서 보다 효율적인 데이터 처리를 지원한다.

o. 관리자 페이지 부여

안드로이드 앱에서 서버 전송에 있어서 데이터 전송과 데이터 관리에 있어 미흡했던 부

분을 보완하는 부분이다. 데이터에 대한 별다른 관리자 페이지가 없었던 부분을 보완하고 최소의 데이터 관리에 있어 효율성을 부여한다.

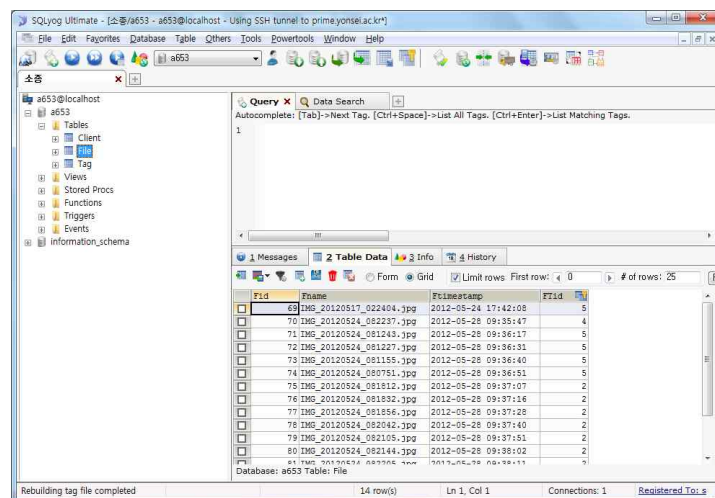
o. 다중 DPF 연결하여 데이터 처리

1,2번의 내용을 바탕으로 다중 DPF에 데이터를 분류하여 출력을 할 수 있도록 지원한다. 이는 다중 DPF에 데이터 분류 출력이 필요한 부분에 사용자 에게 보다 편리한 방향을 제시할 것이다.

4. 개발 진행 사항

o. 데이터 분류

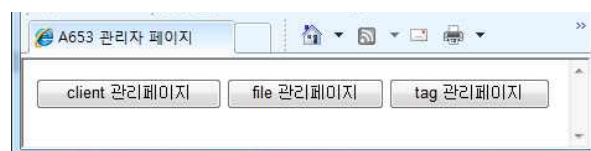
데이터 분류를 위해 DPF서버에 DB를 구축하고 Client, File, Tag 별 Table을 생성하였다. DB의 내용은 안드로이드 application에서 읽기를 하여 사용자들은 서버에서 관리하는 내용으로 태그를 전송 할 수 있도록 지원한다.



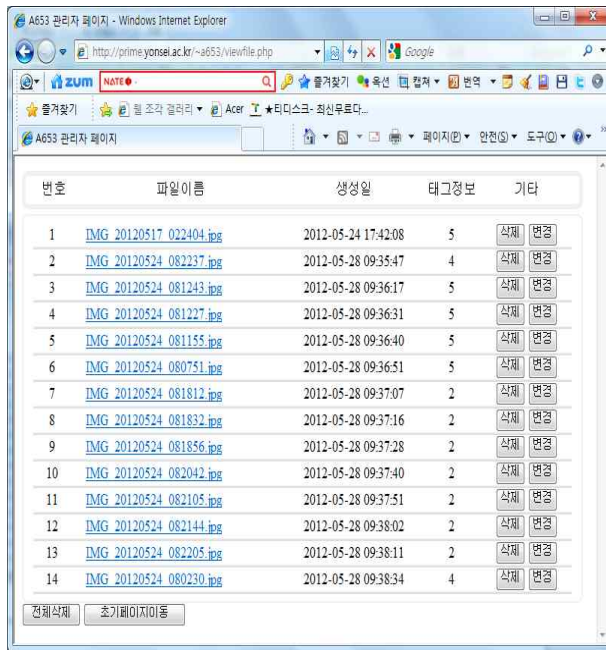
- DB 구축한 내용

o. 관리자 페이지 부여

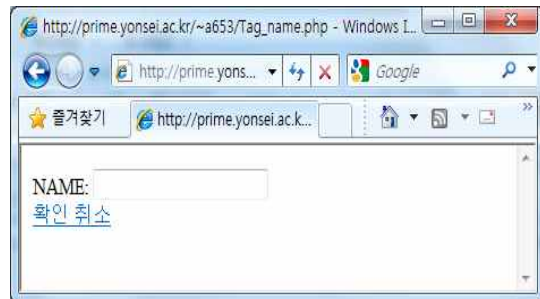
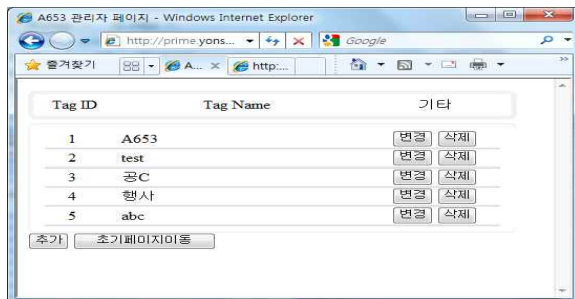
DB구축 이후 서버에서는 데이터 관리에 있어서, 보다 효율적인 방법에 있어 웹 페이지를 구축하였다. 관리자가 수정 할 수 있는 권한은 client와 file에 있어서는 읽기, 수정, 삭제가 가능하며 태그에 대해서는 읽기, 수정, 삭제, 생성이 가능하다.



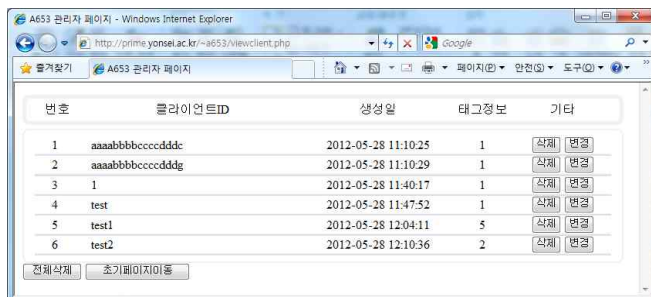
- 관리자 초기페이지 (admin.php)



- 파일 관리 페이지. 태그정보의 변경과 삭제, 읽기가 가능하다. 이미지 이름을 클릭 시 이미지 보기가 가능하다.



- 태그정보 관리 페이지. 태그정보는 추가, 변경과 삭제, 읽기가 가능하다. 추가버튼을 누르면 이름을 입력하고 확인을 눌러 태그를 추가할 수 있다.



- 클라이언트 정보 관리 페이지. 클라이언트 정보 읽기, 태그정보 변경과 클라이언트 삭제, 가 가능하다. 변경버튼을 누르면 현재 관리되고 있는 태그정보들이 출력되며, 태그정보를 선택하여 클라이언트가 가진 태그정보에 대해서 수정을 할 수 있다.

o. 다중 DPF 연결하여 데이터 처리

다중 DPF에 연결하여 데이터 처리를 위해서. 개발한 내용을 android, server, client를 나누어서 기능을 설명한다.

□ Android 기능

1. 앱을 실행하여 사진을 찍고 사진 전송을 누른다.
2. 사진을 선택하고 나면 서버에서 태그정보를 가져와 사용자가 태그를 선택할 수 있도록 한다.
3. 태그를 선택하면 태그정보와 이미지가 서버로 전송되고 선택한 태그를 바탕으로 이미지 정보가 DB에 insert된다.

□ Client 기능

1. Pandaboard를 기반으로 한 리눅스 환경의 클라이언트 프로그램에서 클라이언트 아이디를 통해 서버에 접속한다. 해당 아이디가 담당하는 주제와 관련된 이미지들만을 선별적으로 서버로부터 다운로드한다.
2. 서버에 접속 시, 해당 아이디를 통한 최종접속시간 이후 서버로 전송된 신규 이미지 파일이 있을 시에는 'update' 라는 메시지를 출력하여, 사용자에게 업데이트 사항이 있음을 알리며, 신규 이미지 파일들을 다운로드 받는다.
3. 서버에 접속 시, 해당 아이디를 통한 최종접속시간 이후 서버로 전송된 신규 이미지 파일이 없을 시에는 'no update' 라는 메시지를 출력하여, 사용자에게 업데이트 사항이 없음을 알리며, 아무 동작을 하지 않는다.
4. 해당 기기에서 기존에 사용하였던 클라이언트 아이디와 다른 아이디를 통하여 접속할 경우(id 변경), 기존에 가지고 있던 이미지들을 모두 삭제한 후('delete' 메시지 출력), 새로운 아이디가 담당하는 태그와 관련된 이미지들만을 새로 전송받는다. 이로써, 해당 클라이언트 기기가 담당하고자 하는 주제를 접속 id 변경만을 통하여 자유롭게 선택할 수 있다.
5. 수신된 이미지들을 폴더 안에 관리하며, xslideshow 프로그램을 이용하여 DPF에 이미지들을 슬라이드쇼 형태로 디스플레이 한다.

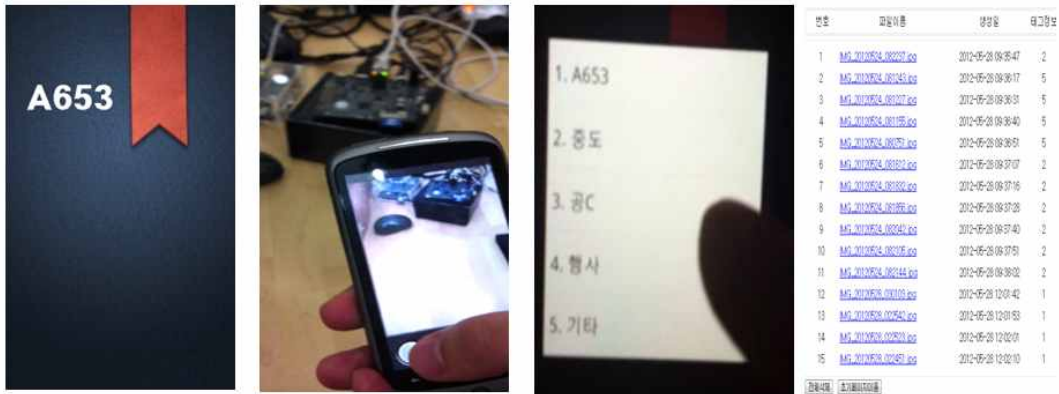
□ Server 기능

1. client가 접속하게 되면 그 정보를 DB에 insert 하여 클라이언트 정보를 관리하게 된다.
2. 수신 받은 이미지에 대해서 선별해주고 클라이언트에게 파일을 전송한다.
3. 태그의 관리를 할 수 있고 추가, 변경, 삭제가 자유롭다.
4. 태그 값을 삭제시 해당 태그값을 가진 이미지는 필요 없게 되므로 자동으로 삭제되게 구현하였다.

5. 구현 결과

학기 초에 프로젝트에 대해서 이야기를 나누고 계획했던 내용들은 모두 개발이 완료 되었다.

태그 전송과 태그 관리



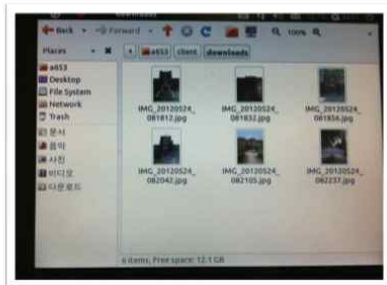
Android image + theme 전송

안드로이드에서 이미지를 전송하는 과정이다.

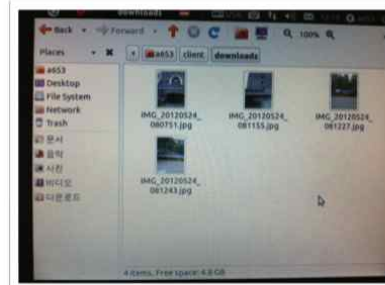
앱을 실행하여 사진을 찍고 DFP SERVER에서 관리하는 태그값을 사용자는 지정할 수 있다. 태그를 전송하면 DB에 태그를 관리하는 자료와 입력과 더불어 파일이 전송이 된다. 이는 관리자페이지의 File 관리에서 확인할 수 있으며, 파일을 클릭 시 해당이미지를 확인할 수 있다. 또한 완성도를 높이기 위해 파일을 삭제 시 DB에 입력된 파일에 해당 태그 값 또한 삭제를 구현하여 DB에 따로 접근 없이 관리자 페이지에서 Data의 관리가 모두 가능하다.

클라이언트별 다중 DPF 연결 구현 결과

Client A



Client B

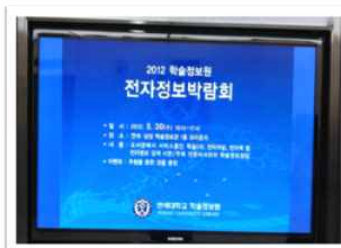


클라이언트별 태그를 지정하고 해당 이미지를 다운받고 출력 할 수 있다.

APPLY

목적에 맞는 이미지 출력

다양한 위치에서 다양한
목적으로 사용



학술정보원

학우들의 참여를 통한
교내 사진 전시등으로
MEMO BOARD 활성화

테마별 이미지

각 테마의 특징적인
실시간 이미지 화면

