|  |  |
| --- | --- |
| 본인평가  등급 | **B** |

**대외개발활동 내역서**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **프로젝트 명** | I’m Your Memory | **주관 기관**  **(회사/단체명/개인**) | | 삼성소프트웨어 멤버십 | |
| **프로젝트**  **담당자** | 김대영 | **시작**  **년월** | 2014.08 | **종료**  **년월** | 2014.09 |
| **개발환경** | OS : Android Kitkat, Window  Language : Java, C++  Tools : Eclipse, Mysql | | | | |
| **프로젝트**  **소개** | USB메모리는 디스켓 이후에 통신을 이용하지 않고 컴퓨터에서 컴퓨터로 데이터를 이동할 수 있는 방법 중에 가장 흔히 쓰이는 방법이다. 디스켓이 없어진 이유는 얇고 그리 무겁지는 않지만 넙적하여 보관시에 주머니에 들어갈 수 있는 아주 간편한 사이즈는 아니라는 것과 USB메모리의 용량이 예전 디스켓에 비해 수천 배에서 수만 배 크기 때문이다. 일반적인 전송방법이 디스켓을 이용하는 것에서 USB메모리를 이용하는 것으로 바뀌었듯이 이제는 USB메모리에서 좀 더 편한 다른 수단으로 바뀌어야 한다고 생각한다. 그래서 현시대에 잘 구축되어있는 모바일 통신망과 WIFI를 이용하여 파일 전송을 하되, USB메모리 만큼 간편하게 사용하도록 하여 사용자의 접근성과 편의성을 높이는 것이 목적이다. 기존의 방법인 USB메모리를 컴퓨터의 USB포트에 꽂아서 바로 전송이 가능한 상태가 되는 것만큼 간단한 방법으로 컴퓨터와 스마트폰이 연결되고 USB메모리같이 윈도우탐색기에서 확인 가능하며 여러 스마트폰을 동시에 연결하더라도 USB메모리를 여러 개 꽂은 듯, 윈도우탐색기를 이용하여 전송이 가능하다. 기존에 나와 있는 비슷한 방법의 애플리케이션들은 IP를 입력하거나, 키 값을 넣거나, 아이디와 비밀번호를 넣어야만 접근할 수 있다. 그러한 방법보다 좀 더 편리한 방법을 제안함으로써 많은 사람들이 데이터케이블과 USB메모리로부터 해방되기를 기대한다. | | | | |
| **본인**  **개발내용** | - Android FTP Server 개발  - PC와 Android간의 매핑을 위한 Server개발 | | | | |
| **참여동기** | 평소 학교에서나 다른 장소에서 필요한 데이터를 접근하기 위하여 USB를 많이 사용하였던 우리는 USB를 자주 잃어버리는 경험을 해보았다 그리고 비교적 대용량을 보유하고 있는 스마트 폰은 데이터케이블이 없는 한 어디서든 접근할 수 없어 이러한 문제점을 해결해보고자 시작하게 되었다. | | | | |
| **프로젝트**  **결과** | Window App Architecture  C:\Users\KTS\AppData\Local\Temp\hunclip1\02\huntemp.files\img0001.jpg  윈도우 애플리케이션은 C++로 제작되었으며 Tray icon의 형태로 백그라운드 프로세스로 항시 동작한다. 스마트폰과의 연동 과정에서 첫 번째 단계에서는 Client의 입장으로서 IP address를 서버에 보내는 작업을 수행한다. 이후 기기간 접촉이 이루어지면 SNTP를 이용하여 타임 서버로부터 접촉 당시의 시간 정보를 표준시간으로 얻어 온다. 동기화 된 시간은 외부 서버에 보내지게 되며, 서버의 매칭 작업이 정상적으로 이루어지면, 스마트폰의 정보를 서버로부터 받게 된다. 윈도우 애플리케이션은 항시 마우스 이벤트를 감지하는데, 마우스 포인터가 데스크탑 우측 상단의 구석에서 2초 가량 머물게 되면 스마트폰의 연결 상태를 슬라이드 효과와 함께 화면에 나타내게 된다. 사용자는 이러한  정보와 함께 제공되는 UI를 통해 손쉽게 스마트폰 메모리에 접근할 수 있다.  Android App Archtecture  C:\Users\KTS\AppData\Local\Temp\hunclip1\04\huntemp.files\img0001.jpg  안드로이드는 크게 자세 제어와 연결 관리 부분으로 나뉘어 진다. 자세 제어는 PC 키보드의 스페이스 바와 접촉할 때 활용이 되며, 스마트폰의 움직임이 단순히 의미없는 움직임인지 아니면 PC와의 접촉을 위한 움직임인지를 구분한다. 연결 관리 부분은 크게 시간 동기화와 외부 서버와의 통신 그리고 SwiFTP 서버가 있다. 시간 동기화는 윈도우 애플리케이션과 마찬가지로 SNTP를 통해 이루어지며 접촉된 시간은 표준시간으로 동기화되어 외부 서버로 보내진다. 매칭이 이루어지면 수초 내 PC로부터 접근 메시지 확인이 가능하다. 확인이 이루어지면 FTP서버를 열고, 스마트폰의 전화번호와 IP Address, 그리고 포트정보가 PC로 넘어가게 되며 본격적인 통신이 시작된다.  IP Mapping Server  C:\Users\KTS\AppData\Local\Temp\hunclip1\06\huntemp.files\img0001.jpg  외부서버로는 통신 부분과 매칭 부분으로 나누어지며, 좀더 세부적으로 보게 되면  통신에서는 Send/ Receive, 매칭에서는 Phone스레드 리스트, PC스레드 리스트 등으로 나누어 설계되었다. 먼저 서버가 시작되면 포트를 열고 클라이언트 연결요청에 대기하게 된다. 클라이언트의 연결 요청이 들어올 경우 서버는 그것을 허용하고 통신을 위한 스트림을 생성한 뒤 클라이언트가 보내 온 패킷을 읽고 처리 할 수 있게 된다. 패킷이 들어오면 보내 온 패킷이 PC의 것인지 스마트폰의 것인지 구분한다. 이후 PC와 스마트폰을 각각 스레드로 나누어 패킷을 송수신하게 되는데 이러한 이유는 PC와 스마트폰의 구분을 통해 좀 더 효율적 클라이언트 관리 및 매칭이 가능해지기 때문이다. 매칭시스템은 스마트폰 스레드에서 Input Time Packet을 수신하게 되면, PC스레드 리스트의 시간들과 비교 잡업을 거쳐 매칭하며, 매칭이 이루어질 경우 서버의 응답에 따라 FTP주소를 PC에게 전송하거나 아니면 새롭게 다시 연결을 시도한다. | | | | |

* 간단한 이미지를 추가하여 시스템 아키텍쳐, 구현 모습 등 설명 가능함.