



국민대학교
전자정보통신대학
컴퓨터공학부


캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	작은 서버
팀 명	원숭이띠 미혼남
문서 제목	계획서

Version	1.6
Date	2018-MAR-09

팀원	강현구 (조장)
	박주언
	송민석
	조경문

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	작은 서버	
	팀 명	원송이띠 미혼남	
	Confidential Restricted	Version 1.6	2018-MAR-09


CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인Ⅰ 수강 학생 중 프로젝트 “작은서버”를 수행하는 팀 “원송이띠 미혼남”의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 “원송이띠 미혼남”의 팀원들의 서면 허락 없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역


Filename	계획서-작은서버.docx
원안작성자	강현구, 박주언, 송민석, 조경문
수정작업자	강현구, 박주언, 송민석, 조경문

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2018-03-05	강현구	1.0	최초 작성	초안 작성
2018-03-07	송민석	1.1	내용 수정	내용 수정 및 보완
2018-03-08	조경문	1.2	내용 수정	내용 수정 및 보완, 병합
2018-03-09	박주언	1.3	내용 수정	내용 수정 및 보완, 병합
2018-03-09	송민석	1.4	내용 수정	내용 수정 및 보완, 병합
2018-03-09	조경문	1.5	내용 수정	내용 수정 및 보완, 병합
2018-03-09	송민석	1.6	최종 작성	내용 수정 및 보완, 병합

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	작은 서버	
	팀 명	원송이띠 미혼남	
	Confidential Restricted	Version 1.6	2018-MAR-09

목 차

1	개요	3
1.1	프로젝트 개요	4
1.2	추진 배경 및 필요성	4
2	개발 목표 및 내용	4
2.1	목표	5
2.2	연구/개발 내용	6
2.3	개발 결과	8
2.3.1	시스템 기능 요구사항	8
2.3.2	비기능(품질) 요구사항	9
2.3.3	시스템 구조	9
2.3.4	결과물 목록 및 상세 사양	10
2.4	기대효과 및 활용방안	11
3	배경 기술	12
3.1	기술적 요구사항	12
3.2	현실적 제한 요소 및 그 해결 방안	13
3.2.1	하드웨어	13
3.2.2	소프트웨어	13
4	프로젝트 팀 구성 및 역할 분담	14
5	프로젝트 비용	15
6	개발 일정 및 자원 관리	16
6.1	개발 일정	16
6.2	일정별 주요 산출물	16
6.3	인력자원 투입계획	17
6.4	비 인적자원 투입계획	17
7	참고 문헌	18

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	작은 서버	
	팀 명	원송이띠 미혼남	
	Confidential Restricted	Version 1.6	2018-MAR-09

1 개요

1.1 프로젝트 개요

‘작은 서버’는 Cellular or Wi-Fi 등 인터넷과 연결되지 않는 상황에서도, 하나의 기기가 서버의 역할을 하게 함으로써, 와이파이를 이용해 근처에 있는 기기와 신호를 공유하여 작은 네트워크를 형성한다.

서버 역할의 AP가 되는 기기의 사용자는 메인 서버와 연결된 작은 서버 서비스 앱에서 서버로 열고 싶은 서비스 콘텐츠를 선택하여 다운로드 받을 수 있고, 이를 통해 서버를 열어 콘텐츠를 공유한다.

1.2 추진 배경 및 필요성

요즈음 우린 인터넷이 없으면 아무것도 못하는 세상에서 살고 있다고 봐도 과언이 아니다. 아침에 눈을 뜨면서, 핸드폰을 손에 쥔 채 인터넷과의 끊임없는 연결은 시작되며 잠자리에 들 때까지 인터넷과 함께하고 있다.


다른 사람들과 대화를 주고 받을 때에도 메신저를, 무료한 시간을 때우기 위해서 네트워크 게임을, 지도를 볼 때, 원하는 정보를 찾을 때 모든 것은 인터넷을 통해 이루어지고, 이것은 너무나 익숙해진 우리의 삶이다.

만약 그 당연하게만 느껴지는 인터넷의 연결이 되지 않는 상황이라면, 당신의 스마트 기기들은 세상과의 커뮤니케이션이 제한된 매력 없는 고철 덩어리일 뿐이다.

그 상황이 다른 사람들과의 소통이 절실히 필요한 순간이라면 상황은 심각해진다. 당신은 조난당했다. 통화권 이탈지역이다. 당연히 인터넷과 연결되지 않는다. 아무도 당신이 그곳에 있는지 알지 못한다. 그대로 하염없이 누군가 구해주기를 기다릴 수 밖에 없다.

그 때, 구조대원과의 연결이 가능하다면 어떨까?
아무런 통화 신호가 잡히지 않아도, 당신은 조난자를 찾고 있는 구조대원에게 당신이 살아있음을 알린다.

‘작은 서버’를 이용해서 말이다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	작은 서버	
	팀 명	원송이띠 미혼남	
	Confidential Restricted	Version 1.6	2018-MAR-09

비단 극단적인 재난상황에만 국한되는 것이 아니다.
비행기 기내처럼, 인터넷 사용이 금지된 공간에서도 작은 서버가 활약할 수 있다.
장시간의 비행은 지루하기 그지 없다. 항공사가 제공하는 미디어, 엔터테인먼트 서비스는 긴 시간을 달래기에는 부족하다.
물론, 핸드폰에 미리 받아 놓은 영상을 보거나, 음악을 감상하거나, 책을 읽을 수 있다.
개인적으로 할 수 있는 것들은 많다. 하지만 혼자 하면 무엇이든 질리기 마련, 핸드폰을 통해 누군가와 소통하며 간단한 채팅, 혹은 게임을 즐길 수 있으면 한결 비행이 즐거워질 것이다.
비행기에서 승객들끼리 만날 수 있는 온라인 공간을 만들어 줄 수 있다면 어떨까?

한 사람이 자신의 핸드폰에서 게임 서버를 열고, AP가 되어 Wi-Fi신호를 수신 가능한 주변 사람들과 게임을 공유하여 플레이할 수 있다. 그들만의 작은 놀이터가 만들어지는 것이다.


2 개발 목표 및 내용

2.1 목표

본 프로젝트는 서버 플랫폼으로서 추가적인 어플리케이션 설치 없이도 서버 플랫폼 어플리케이션을 설치하였고, AP 역할을 하는 유저에게 Wi-Fi를 통하여 접속해서 우리 서비스가 제공하는 기능들을 이용 할 수 있게 개발하는 것을 목표로 한다.

우리 서비스를 사용하는 유저가 Contents Developer로서 콘텐츠를 업로드, 병합 할 수 있게 하고, local server로서 서버 다운로드, 삭제, 실행 할 수 있게 한다. 또한 1대의 디바이스를 이용해 서비스를 사용하지 않는 유저들도 우리의 기능을 이용 할 수 있게 하는 것을 목표로 한다.

본 프로젝트가 완성되었을 때 local server들을 통해 우리 서비스를 이용하지 않는 유저들도 서비스의 기능들을 이용 할 수 있게 하거나, ad hoc network를 이용하여 local server들을 연결 시켜 서버를 구축하는 것을 목표로 한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	작은 서버	
	팀 명	원송이띠 미혼남	
	Confidential Restricted	Version 1.6	2018-MAR-09

2.2 연구/개발 내용

우리 서버 플랫폼은 네트워크가 없는 장소에서도 우리가 제공하는 서비스를 이용 할 수 있다는 것이 핵심이다. 우리 서비스는 local server 들을 이용하여 네트워크가 없는 장소나 특정 지역에서 본인이 원하는 기능들을 APP 설치 없이도 사용 할 수 있다. 유저가 Wi-Fi 를 hotspot 을 터트려 rooting 이 된 디바이스로 AP 를 구축한다. Application 을 사용하고 있는 유저가 device 를 AP 로 사용하면 다른 유저들이 이를 Wi-Fi 로 잡아 Application 설치 없이도 유저가 사용하는 Application 의 기능들을 이용 할 수 있다. 이는 Web or raspberry pi 에서도 이용 가능하여 활용 방안이 크다. Application 의 main server 를 이용하게 되면 즉 네트워크를 사용하여 접속하면 모든 유저들이 올려놓은 여러 기능들을 다운 받아 이용 할 수 있게 되고, 유저들이 개발자가 되어 우리 서비스에 좋은 기능들을 올릴 수 있게 툴을 제공할 예정이다. 우리 서비스는 ad hoc network 를 이용하여 local server 들을 연결 시켜 서버를 구축한다. 오늘날 무선 네트워크의 경우, 노드와 사용자 이동성은 주로 포워딩을 통해 처리된다. 하지만 이 같은 포워딩 방식은, 네트워크 가장자리에 있는 노드(단말기)가 움직일 때만 적용되기 때문에 ad hoc 네트워크처럼 네트워크 중앙에 위치한 노드가 움직이거나 통신 기기가 라우터와 호스트 기능을 동시에 수행하는 경우에는 불가능하다. 따라서 ad hoc 네트워크의 경우, 라우팅 알고리즘이 이동성을 직접 처리한다. 만약 노드가 움직여 트래픽을 다른 쪽으로 강제로 옮기면, 라우팅 프로토콜은 노드의 라우팅 테이블에 일어난 변화를 관리한다.

- AP 로 동작하는 서버 기기와 클라이언트로서 동작하는 기기와 연결


서버를 인터넷 연결이 되어있지 않은 안드로이드 기기에서 실행하고, 테더링 기능을 활성화하게 만든다.

현재 안드로이드 환경에서는 수동으로 모바일 Hotspot 을 켤 수 있는 기능이 존재하나, 해당 프로젝트의 시스템을 사용하기 위해 수동으로 Hotspot 을 키는 방식은 번거로울 뿐 더러, 안드로이드 운영체제에서 기능적으로 에어 플레인 모드에서는 Hotspot 을 동작할 수 없는 등의 제약 사항이 따르므로, Hotspot 기능을 강제적이며 자동적으로 활성화하는 기능을 해당 시스템에 추가한다.

클라이언트 유저는 해당 시스템 앱을 거치지 않고 브라우저를 통해 로컬 서버에 접근할 수 있게 한다.

-컨텐츠 개발자의 메인 서버로의 컨텐츠 업로드

컨텐츠 개발자는 자신이 만든 컨텐츠를 AWS EC2 기반의 기 구축되어 있는 서버에 복잡한 절차 없이 업로드 할 수 있게 만든다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	작은 서버	
	팀 명	원숭이띠 미혼남	
	Confidential Restricted	Version 1.6	2018-MAR-09

채팅, 게임 등의 기능을 수반하는 콘텐츠를 개발한 후, 해당 시스템의 앱 내 '콘텐츠 병합' 기능을 이용해 해당 콘텐츠를 서버로 동작하게 하는 module 과 병합한다.

이와 같은 module 은 개발자들의 콘텐츠가 어느 것인지와 상관없이 호환이 정상적으로 이루어져야 한다. 그 후 '콘텐츠 업로드' 기능을 통해 해당 콘텐츠서버를 메인 서버에 업로드하게 된다.


-콘텐츠 Server 를 만들기 위한 App 사용자의 콘텐츠 다운로드

App 을 사용하는 사용자는 앱을 통해 메인 서버로부터 원하는 콘텐츠를 다운받는다. 다운 완료된 콘텐츠는 실행 또는 삭제를 할 수 있다. 사용자의 기기가 local server 로서의 기능을 하며 콘텐츠를 제공한다.

- AP 로 동작하는 서버 기기와 AP 로 동작하는 서버 기기와의 연결

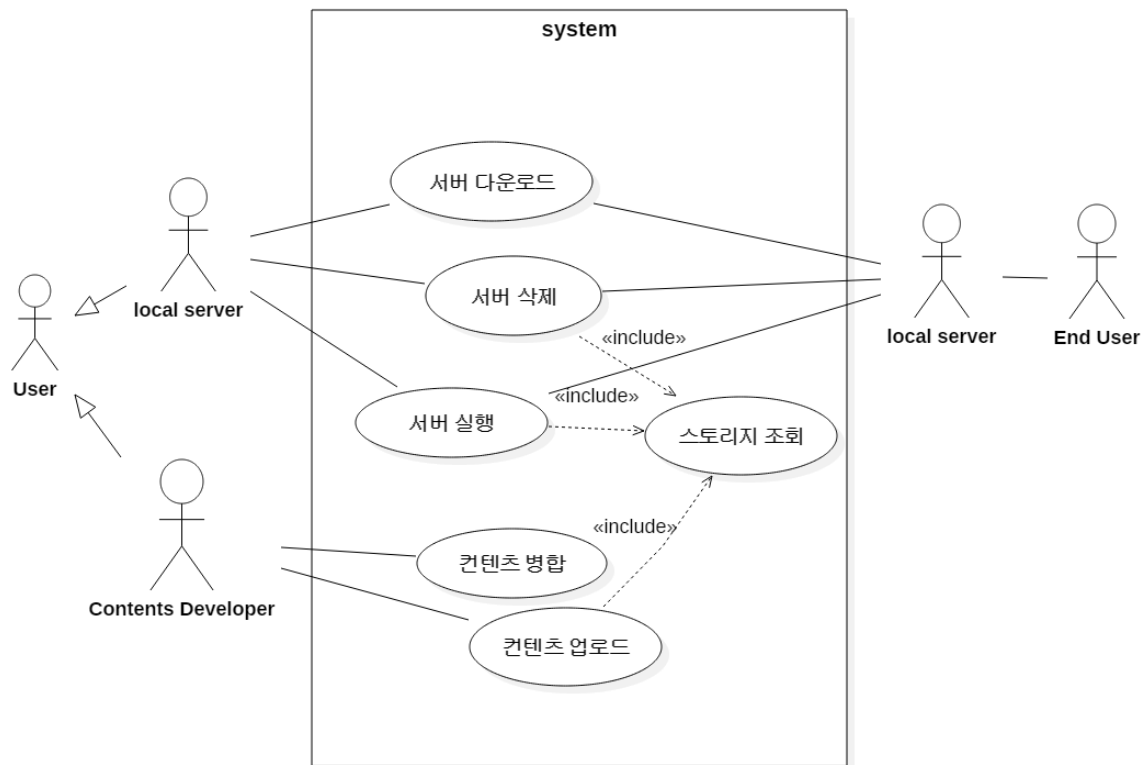
현재 안드로이드 환경에서는 Hotspot 을 켜 놓은 상태의 기기에서의 Wi-Fi 연결을 지원하지 않는다. 따라서, Hotspot 기능과 Wi-Fi 모드를 함께 활성화하는 것을 강제할 수 있는 기능을 해당 시스템에 추가한다. 해당 기능은 앱 내에서 수동으로 선택할 수 있다.

서버 유저는 다른 서버에 연결되며, 이를 통해 서버들의 연쇄적인 연결이 가능하다. 작은 서버가 모여 Ad hoc 방식의 네트워크를 형성할 수 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	작은 서버	
	팀 명	원송이띠 미혼남	
	Confidential Restricted	Version 1.6	2018-MAR-09

2.3 개발 결과

2.3.1 시스템 기능 요구사항




local Server: App을 이용해 서버를 가동하는 유저

Contents Developer: 콘텐츠를 만들고 App을 통해 메인 서버에 업로드하는 개발자

End User: local Server 유저가 만든 서버와 연결되는 클라이언트 유저

- 서버 다운로드: App을 통해 메인 서버와 연결 후, 원하는 콘텐츠 서버를 선택 후 다운로드한다.
- 서버 삭제: App을 통해 스토리지에 접속 후, 목록에서 원하는 콘텐츠 서버 항목을 삭제한다.
- 서버 실행: App을 통해 스토리지에 접속 후, 목록에서 원하는 콘텐츠 서버 항목을 실행한다.
- 스토리지 조회: App을 통해 스토리지 내의 콘텐츠 서버 목록을 불러온다.
- 콘텐츠 병합: App에서 콘텐츠 개발자가 만든 콘텐츠를 서버가동을 위한 모듈과 병합한다
- 콘텐츠 업로드: App을 통해 스토리지에 접속 후, 목록에 원하는 콘텐츠 항목을 메인 서버에 업로드 한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	작은 서버	
	팀 명	원송이띠 미혼남	
	Confidential Restricted	Version 1.6	2018-MAR-09

2.3.2 시스템 비기능(품질) 요구사항

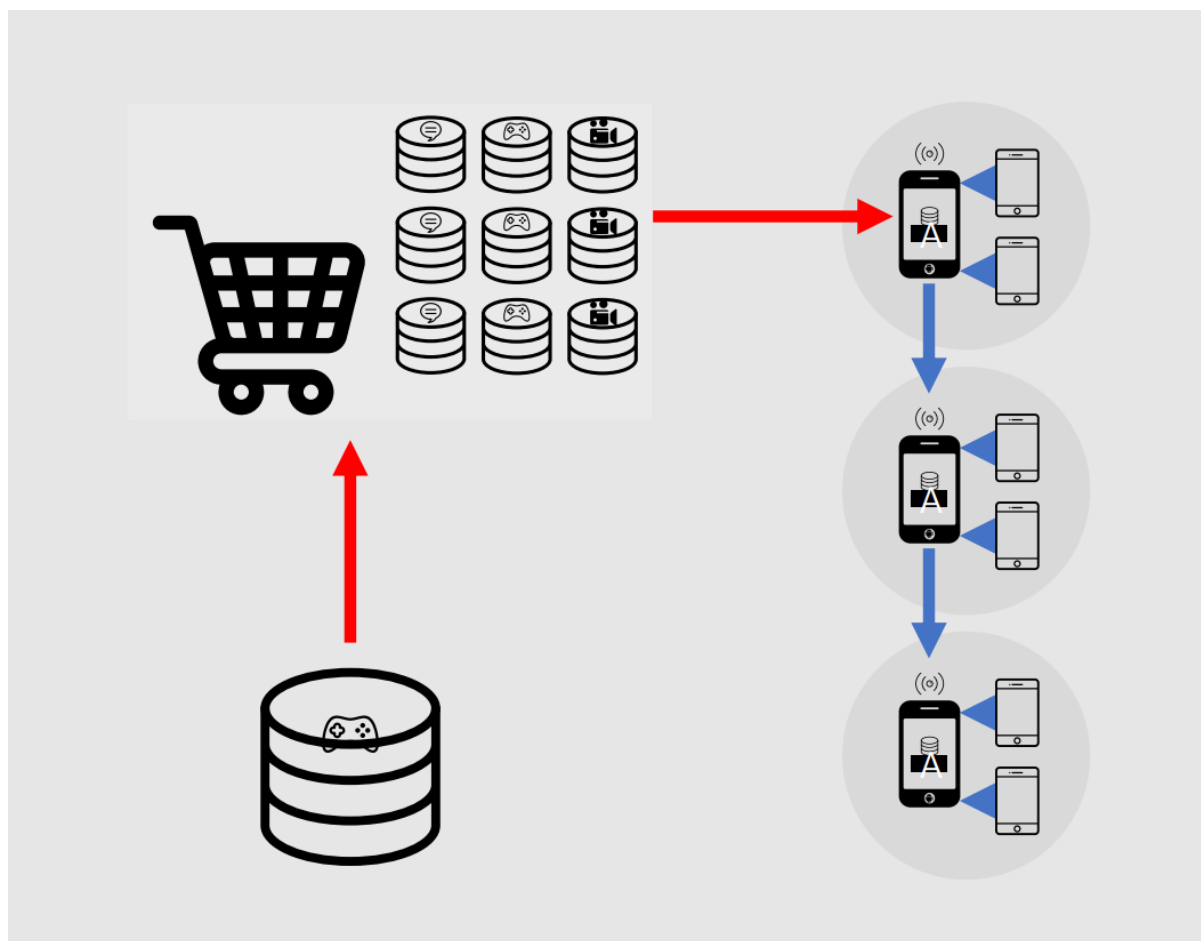
거리적으로는 꽤 먼 거리를 커버하고, 속도적으로는 공유하는 콘텐츠에 딜레이가 걸리지 않도록 만들어야 한다.


블루투스나 NFC보다는 속도적으로나 거리적으로나 WIFI를 이용하는게 무난하다고 생각되어 WIFI로 구현.

하드웨어적 품질 요구사항에서 중요할 수 있는 부분이 배터리 이슈라고 생각한다.

차후 발전할 배터리적 성능을 생각해 큰 문제가 될 거라고 여기진 않았다.


2.3.3 시스템 구조



 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	작은 서버	
	팀 명	원송이띠 미혼남	
	Confidential Restricted	Version 1.6	2018-MAR-09

2.3.4 결과물 목록 및 상세 사양

대분류	소분류	기능	형식	비고
안드로이드 App	컨텐츠	컨텐츠 업로드, 다운로드, 삭제, 실행		
	컨텐츠 테스트	컨텐츠 테스트		
raspberry pi	컨텐츠	컨텐츠 업로드, 다운로드, 삭제, 실행		
	컨텐츠 테스트	컨텐츠 테스트		
Web	컨텐츠	컨텐츠 업로드, 다운로드, 삭제, 실행		
	컨텐츠 테스트	컨텐츠 테스트		
서버	메인 서버	메인 서버 구성		
		메인 서버와 앱 연결 구성		
	작은 서버	작은 서버 구성		
		작은 서버끼리 연결 구성		
		작은 서버와 end user 연결 구성		
	랭킹 서버	랭킹 서버 구성		
컨텐츠	Tool	개발자 Tool 구성		

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	작은 서버	
	팀 명	원송이띠 미혼남	
	Confidential Restricted	Version 1.6	2018-MAR-09

2.4 기대효과 및 활용방안

현재 대부분 휴대전화 사용자들은 Cellular or Wi-Fi 를 통한 인터넷 연결이 없으면 오프라인을 통한 개인활동(음악듣기, 다운 받아 놓은 콘텐츠 즐기기) 이외에는 휴대전화를 통해 할 수 있는게 없다고 봐도 무방했다.

하지만 누군가 자신의 휴대전화를 통해 AP 역할을 하거나 raspberry pi 가 AP 역할을 한다면 인터넷 연결 없이도 주변 사람들과, 혹은 이어지고 이어져서 더 넓은 지역, 많은 사람과의 소통이나 공유가 가능해질 수 있게 될 것이다.

우리가 만든 앱이 가장 활약할 수 있는 부분은 큰 재난이나 예기치 못했던 어떤 일로 인해 인터넷 망이 모두 끊긴 상황이 생기고, 사람들을 구출해야 할 상황일 때, 실종자 중 누군가, 혹은 많은 구조대원들이 우리의 앱을 통해 AP 가 되고, 실종자들이 이곳에 접속하여 위치나 주변 정보 등을 공유한다면 구조 진행속도가 비약적으로 빨라질 것이라고 믿어 의심치 않는다.

위같이 극단적인 상황이 아닌 경우에도 인터넷 연결이 불가능한 비행기 내에서 장시간 비행이라면 아무래도 함께 AP 에 접속하여(언급한 AP 는 비행기에 미리 설치된 raspberry pi 가 될 수도, 앱을 설치해 놓은 심심한 개인이 될 수도 있을 것이다) 대화, 혹은 간단한 게임이라도 즐긴다면 혼자서 시간을 보내는 것보다 훨씬 더 질 높은 시간이 될 수 있을 것이라 생각한다.

위 예시 외에도 인터넷이 안되는 곳 어디든 이 앱을 이용한다면 비록 작은 공간이라고 생각 할 수도 있지만, 인터넷의 축소판과 같은 작은 서버가 생기고 APP 속의 콘텐츠들을 통해 무궁무진한 일들을 할 수 있을 것이다.

인터넷이 되는 곳에서도 활용방안을 찾아 볼 수 있다.

간단한 예를 들면 대학 수업시간에 교수가 문제를 올려놓은 서버를 터뜨리고, 학생들이 이에 접속하여 문제들을 풀어보고 결과를 제출 하는 활동을 할 수도 있을 것이고, 대기시간이 매우 긴 레스토랑에서 앱을 통해 간단한 게임서버를 터뜨려 제공한다면 기다리는 고객들의 화를 누그러뜨릴 수도 있을 것이다.

대기업에서 기밀회의를 할 일이 있을 때 앱을 통해 회의를 주최하는 사람이 Cellular, Wi-Fi 를 모두 끈 후 우리 앱을 통해 회의를 한다면 회의 내용이 참석자들 밖으로는 빠져나갈 수 없다는 건 사용자들도 알 수 있을 것이며, 기밀성이 보장 될 수 있다는 장점이 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	작은 서버	
	팀 명	원숭이띠 미혼남	
	Confidential Restricted	Version 1.6	2018-MAR-09

3 배경 기술

3.1 기술적 요구사항

* 테더링: Wireless-LAN(Wi-Fi) 기술을 이용한 Private Hotspot 셰어링 기능

3.1.1 개발환경

1) 어플리케이션

- 개발 OS: WINDOWS 10, Linux, MAC OS X
- 개발 언어: JavaScript, Python, HTML, CSS, JAVA, Node.js, PHP
- 개발 환경: Android Studio, Brackets, eclipse, Nginx, raspberry pi

2) 서버(메인)

- 서버 환경: AWS EC2

3.1.2 결과물 확인환경

Android OS, Raspberry Pi (Linux 기반)

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	작은 서버	
	팀 명	원송이띠 미혼남	
	Confidential Restricted	Version 1.6	2018-MAR-09


3.2 현실적 제한 요소 및 그 해결 방안

3.2.1 하드웨어

- * 서버를 구동하는 한 기기에 여러 대의 클라이언트 기기들이 연결되면, 서버를 구동하는 기기에 과부하가 걸린다. 스마트폰일 경우에는 배터리소모와 발열이 극심해질 우려가 있다.
- 해결방안: 주기적인 다중 접속이 잦을 것으로 예상되는 장소에는 여러 클라이언트의 요청을 수용할 수 있는, 전원이 연결된 서버전용기기를 설치하는 방안이 있다. (예) 기내)

3.2.2 소프트웨어

- * 안드로이드 환경에서는 에어 플레인 모드에서 Hotspot을 키는 것을 제한한다.
- 해결방안: rooting을 한 안드로이드 기기 혹은 기타 앱을 통해서도 해당 기능을 활성화할 수 있는 것으로 보아, 기술적으로 불가능한 것이 아님을 알 수 있다. 강제로 제어하는 방법을 모색한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	작은 서버	
	팀 명	원숭이띠 미혼남	
	Confidential Restricted	Version 1.6	2018-MAR-09


4 프로젝트 팀 구성 및 역할 분담

이름	역할
강현구	<ul style="list-style-type: none"> - 메인 서버 구현 - 개발자 tool 구현 - 개발자 <-> 메인 서버 구현 (컨텐츠 테스트 및 upload 기능 제공)
송민석	<ul style="list-style-type: none"> - 컨텐츠 개발 - 메인 App 구현(UI) - Web page 구성
박주언	<ul style="list-style-type: none"> - end user <-> 작은 서버 연결 구현 - end user 가 접속할 웹 구현 - 메인 서버 -> 앱 구현
조경문	<ul style="list-style-type: none"> - 메인 App 구현(기능) - raspberry pi (안드로이드처럼) 구현 - Web page 구성

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	작은 서버	
	팀 명	원숭이띠 미혼남	
	Confidential Restricted	Version 1.6	2018-MAR-09

5 프로젝트 비용

항목	예상치 (MD)
관련 기술/오픈소스 검색 및 분석	40
어플리케이션 구성	40
Web 구성	20
raspberry pi 구성	20
작은 서버와 End User 연결 구성	20
작은 `서버끼리 결 구성	40
메인 서버와 앱 연결 구성(컨텐츠 요청 및 전송)	30
메인 서버 구성	20
개발자 Tool 개발	60
개발자 -> 메인 서버 컨텐츠 업로드 기능 구현	30
컨텐츠 테스트 개발	20
랭킹 서버 구성	20
보고서 작성	20
합	380

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	작은 서버	
	팀 명	원송이띠 미혼남	
	Confidential Restricted	Version 1.6	2018-MAR-09


6 개발 일정 및 자원 관리

6.1 개발 일정

항목	세부내용	1 월	2 월	3 월	4 월	5 월	비고
요구사항분석	아이디어 구상						
	주제 선정						
관련분야연구	주요 기술(App, Web, raspberry pi, main server) 연구						
	관련 기술 및 오픈소스 검색 및 분석						
설계	시스템 설계						
구현	end user 와 작은 서버 연결 구현						
	서버와 개발자 연결 구현						
	개발자 tool 개발						
	제공할 콘텐츠 개발						
테스트	시스템 테스트						
최종 발표	발표 준비 및 발표						

6.2 일정 별 주요 산출물

마일스톤	개요	시작일	종료일
계획서 발표	프로젝트 아이디어 선정 프로젝트 자료 수집 산출물 : 1. 프로젝트 수행 계획서 2. 프로젝트 기능 일람표	2018-01-22	2018-03-09
설계 완료	작은 서버 & 엔드 유저 연동 산출물 : 1. 작은 서버와 엔드 유저 연결 구축	2018-03-09	2018-03-20
1 차 중간 보고	메인 서버 & 앱 구축 완료 산출물 : 1. 프로젝트 1 차 중간 보고서 2. 프로젝트 진도 점검표 3. 1 차분 구현 소스 코드	2018-03-21	2018-04-14

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	작은 서버	
	팀 명	원송이띠 미혼남	
	Confidential Restricted	Version 1.6	2018-MAR-09


2 차 중간 보고	컨텐츠 개발자 툴 개발 완료 메인 서버& 앱 연동 완료 산출물 : 1. 프로젝트 2 차 중간 보고서	2018-04-15	2018-05-19
구현 완료	시스템 구현 완료 산출물: 메인 서버와 연동되는 작은 서버	2018-04-15	2018-05-20
테스트	시스템 통합 테스트 산출물: 최종본	2018-05-21	2018-05-25
최종 보고서	최종 보고 산출물: 최종보고서	2018-05-20	2018-05-24

6.3 인력자원 투입계획

이름	개발항목	시작일	종료일	총개발일(MD)
조경문	메인 App, Web 구현	2018-03-07	2018-03-30	
조경문	메인 raspberry pi 구현	2018-03-07	2018-03-30	
박주언, 강현구	메인 서버 구성	2018-03-07	2018-04-01	
박주언	작은 서버와 end user 연결 구현	2018-03-07	2018-03-30	
강현구	메인 서버와 작은 서버 연결 구현	2018-04-01	2018-05-10	
송민석	컨텐츠 만들기	2018-03-07	2018-05-10	
송민석	개발자 tool 만들기 및 메인 서버로 업로드 구현	2018-03-07	2018-04-20	
전체	구현 한 것 연동 후 테스트/ 수정	2018-05-10	2018-05-28	
전체	발표 준비	2018-05-20	2018-05-28	

6.4 비 인적자원 투입계획

항목	Provider	시작일	종료일	Required Options
개인 노트북	레노버, 삼성, 애플	2018-01-09	2018-05-29	
raspberry pi	raspberry pi	2018-03-09	2018-05-29	
갤럭시 휴대전화	삼성	2018-03-09	2018-05-29	

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	계획서		
	프로젝트 명	작은 서버	
	팀 명	원숭이띠 미혼남	
	Confidential Restricted	Version 1.6	2018-MAR-09

7 참고 문헌

번호	종류	제목	출처	발행 년도	저자	기타
1	사이트	https://stackoverflow.com/questions/35325454/create-a-lan-access-point-on-android-device-no-internet-needed/35327467#35327467	스택 오버 플로			우리가 하려는 것 과 관련된 질문
2	사이트	https://academy.realm.io/kr/posts/difference-between-kotlin-java/				kotlin
3	어플	Servers Ultimate	플레 이스 토어			하려는 거랑 연관 된 앱
4	강의	https://www.youtube.com/watch?v=7ThkvfCKKQs&list=PLuHgQVnccGMC5AYnBg8ffg5utOLwEj4fZ&index=1	유튜 브			AWS
5	사이트	http://mobileandlife.tistory.com/entry/ 모바일 기술 정리-와이파이-다이렉트-WIFI-Direct	블로 그			와이파이
6	서적	더 도커 북	루비 페이 퍼	201 4.10 .10	제임 스 턴볼	
7	사이트	https://m.blog.naver.com/PostView.nhn?blogId=horusi&logNo=220337395781&proxyReferer=https%3A%2F%2Fwww.google.co.kr%2F				AP 만들기
8	사이트	http://comterman.tistory.com/621				와이파이 다이렉 트
9	사이트	http://hsj0511.tistory.com/205				ajax
10	사이트	https://ruslanspivak.com/lbaws-part3/#fn-1				웹서버 만들기
11	사이트	https://blog.outsider.ne.kr/312#recentEntries				http method