****

**2016.06.08**

**이규영**

**임세정**

**최지운**

**2016 컴퓨터 시스템 기초 설계**

**최종 설계 보고서**

1. 1. 문제상황(실제 사례) - 타지에서 대중교통파악불능

1)

[구글 지도] [오전 3:18] **네이버지도봐도**

[구글 지도] [오전 3:18] **내가어딨는지모르겠음**

[아이나비] [오전 3:18] 딱 내리는 곳에서 nfc나 qr찍으면

[아이나비] [오전 3:18] 바로 뜨는거

[다음 로드뷰] [오전 3:18] 응응

[아이나비] [오전 3:18] 주변에 머가 있고

[아이나비] [오전 3:18] 딱딱 나오는거

[아이나비] [오전 3:18] 어때

[다음 로드뷰] [오전 3:18] ㅇㅇㅇㅇㅇㅇ

[다음 로드뷰] [오전 3:18] 그럼 되지

[아이나비] [오전 3:18] 오

[다음 로드뷰] [오전 3:18] 아주 좋네요

[아이나비] [오전 3:18] 그래서 내가 지정한 곳으로

[아이나비] [오전 3:18] 이끌어주는거야

[다음 로드뷰] [오전 3:18] 그걸로 밀고 나가!

[아이나비] [오전 3:18] qr코드찍으면서

[다음 로드뷰] [오전 3:18] 응응

2)

[구글 지도] [오전 3:14] **다른지역갔는데**

[구글 지도] [오전 3:14] **대중교통**

[구글 지도] [오전 3:14] **1도몰라서**

[다음 로드뷰] [오전 3:14] 아 맞아

[다음 로드뷰] [오전 3:14] ㅇㅈ

[구글 지도] [오전 3:14] 택시타고다니는거

[다음 로드뷰] [오전 3:14] 그거 졸라 싫어

[파인 드라이브] [오전 3:14] ㅇㅈ

3)

[다음 로드뷰] [오전 3:16] 목포 버스 어플은 위치가 너무 부정확해

[구글 지도] [오전 3:16] 대전버스어플도

[다음 로드뷰] [오전 3:16] 나 그래서 정류장 5개나 더 감

[다음 로드뷰] [오전 3:16] ㅡㅡ

[구글 지도] [오전 3:16] 부정확해

[다음 로드뷰] [오전 3:16] 목포였는데

[다음 로드뷰] [오전 3:16] 졸라 걸어서 한참 걸어갖고

[다음 로드뷰] [오전 3:16] 겨우 감

[구글 지도] [오전 3:17] 그래서 모르는곳은

[구글 지도] [오전 3:17] 나도 대전에서 택시탔어

[다음 로드뷰] [오전 3:17] **버스 방향도 제대로 좀 알랴줬으면 좋겠음**

[다음 로드뷰] [오전 3:17] 1도 모르겠어

[구글 지도] [오전 3:17] 길치를위한

[구글 지도] [오전 3:17] ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ

[다음 로드뷰] [오전 3:17] **이 방향으로 가는 게 맞는 건지 모르겠음**

[다음 로드뷰] [오전 3:17] ㅇㅇㅇ

1. 2. 5Whys

|  |  |  |
| --- | --- | --- |
| 1 | 왜 길을 제대로 가지 못하는가? | 타지는 익숙치 않아서 |
| 2 | 타지는 왜 익숙치 않은가? | 그 곳에 대한 정보가 없기 때문에 |
| 3 | 왜 처음가는 곳에 대해 정보가 없는가? | 평상시 우리가 이용하는 대중교통과는 너무 달라서 정보와는 상관이 없기 때문에 |
| 4 | 왜 우리가 이용하는 대중교통과 다른가? | 지역마다 특성이 다르고 버스 번호가 다르기 때문에 |
| 5 | 어떻게 해결할 것인가? | 정류장 마다 qr코드 혹은 nfc로 내가 가고자 하는 목적지에 대해 그 자리에서 가장 편한 법으로 알려준다. |

1. 3. 필요성
2. 21세기 정보화 시대에 걸 맞는 시간절약과 에너지 절약을 하기 위함
3. 그 누구보다 빠르게 도착하기 위함
4. 누구 보다 빠르게 난 남들과는 다르게 지도 위에 완벽한 나그네가 되기 위함
5. 비싸디 비싼 택시를 타지않음으로써 경제 성장력을 키우기 위함
6. 4. 요구사항
7. 타 지역에 갔을 때 주변 대중교통수단에 대해 알 수 있어야한다.
8. 언제 어디서 어떻게 오는지 한 번에 알고 싶다.
9. 버스가 언제 어느 방면에서 오는지 버스가 정류소에 도착까지 남은 시작을 알고싶다.
10. 지도 어플들과 연동이 되었으면 좋겠다.

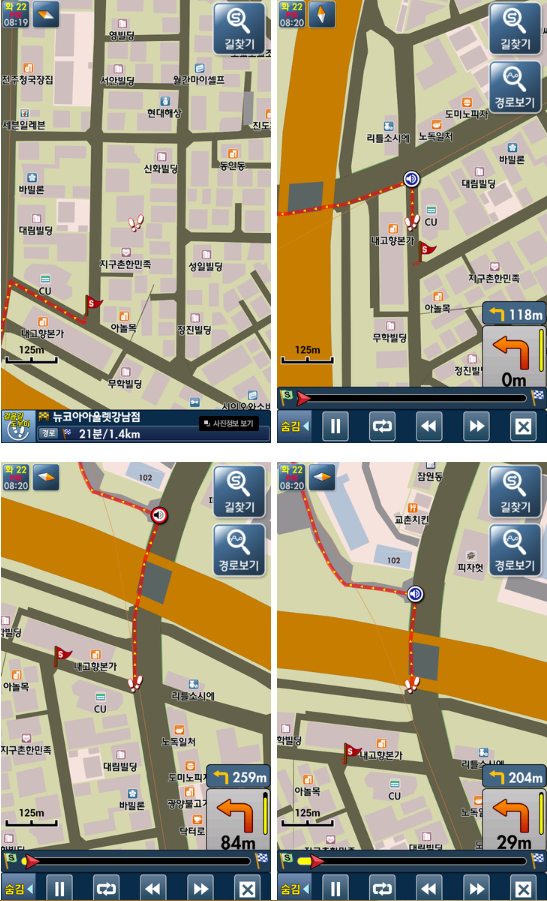
5) 비둘기 보다 실현 가능한 어플이여야 한다.

**5. 해결책 아이디어 도출**

1. 정류장마다 체크포인트(qr코드 or nfc)를 설치하여 현재 상황을 쉽게 파악
2. 지도와 연동 후 지하철과 버스상황을 실시간으로 보여준다.

**6. 유사한 문제 해결 방법**

1. 보행자 네비게이션 – 걸음길 도우미

* **서울 특별시에서만 가능** 하지만 보행자들만을 위한 어플이다. 하지만 대중교통에 대해서는 여느 어플과 같이 가는 경로와 몇 번을 타면 되는지만 알려줄 뿐이다. 그저 사람이 걸어다니는 길로 표현해 놨을 뿐이다.

1. 指さしナビ

* 사진에 첨부한 것과 같이 가는 방향을 손가락으로 가리켜주며 현 나의 위도와 경도를 보여준다. 이는 매우 편리한 기능이다. 하지만 일본 일정 지역만 사용 가능하며 가독성이 떨어진다. 나는 그저 손가락이 가리키는 대로 폰만 보며 걸어가야 하고 데이터를 계속 켜놓은 상태로 다녀야 하기에 **데이터 부담**이 심히 걱정된다.



**7. 시스템의 구성 요소 및 기능 구성**

1. 시스템의 구성 요소

* QR코드 or NFC
* GPS
* WIFI/셀룰러 데이터
* App

1. 기능 구성

* QR코드 or NFC : 정류소에 체크포인트, 즉 쉽게 얘기하자면 스탬프 투어처럼 핸드폰을 NFC 기능을 이용해 장소를 인식시키면 주변 상황에 대하여 알려줍니다. 주변 대중교통의 현 상황을 실시간으로 알려줍니다. **올바르게 가고 있는지 중간점검**을 하는

역할을 합니다.

* GPS & WIFI : GPS를 통하여 현재 나의 위치를 알 수 있고 WIFI를 통하여 더욱 정확한 위치 인식이 가능합니다.
* App : (앱의 기능을 부분화 및 세분화)
  + NFC 데이터 송수신 대중교통 실시간 정보+노선도
  + Gps위치 송수신 – 자신의 현 위치를 좌표로 받는다.
  + 스크린 – gps를 기반 지도에 사용자의 위치를 출력함
  + 최적의 루트 계산 – 버스대기시간 + 버스탑승시간이 가장 적은 루트로 지정하는 알고리즘

(+권한 획득)

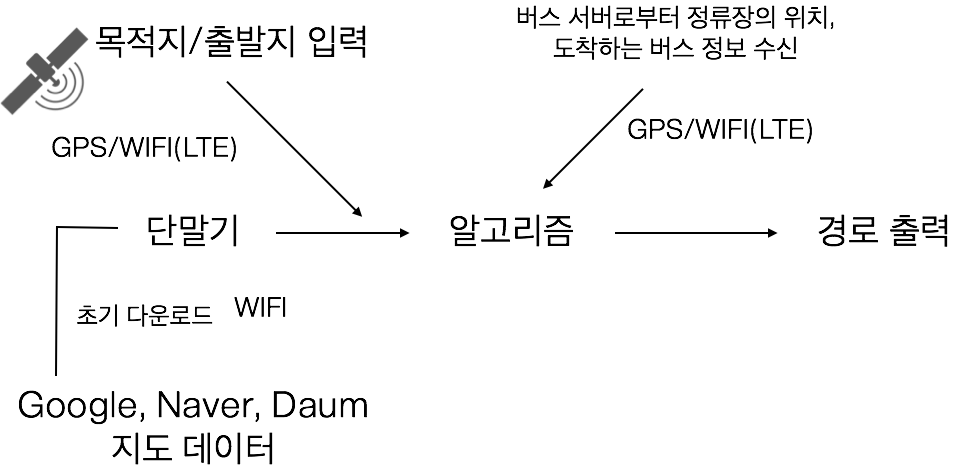
-참조-

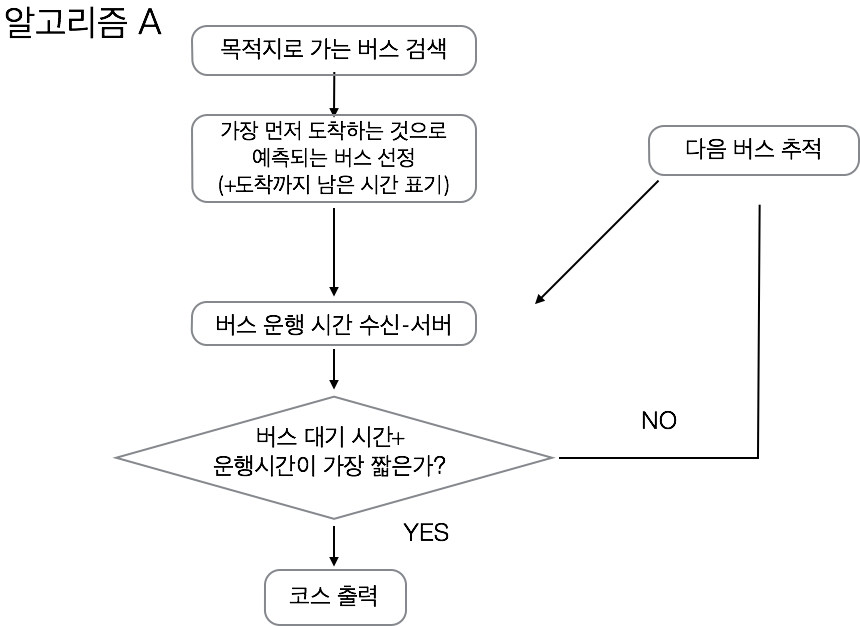
* + 구글지도-데이터베이스
  + 네이버지도-인터페이스
  + 나침반 – 동서남북 표기

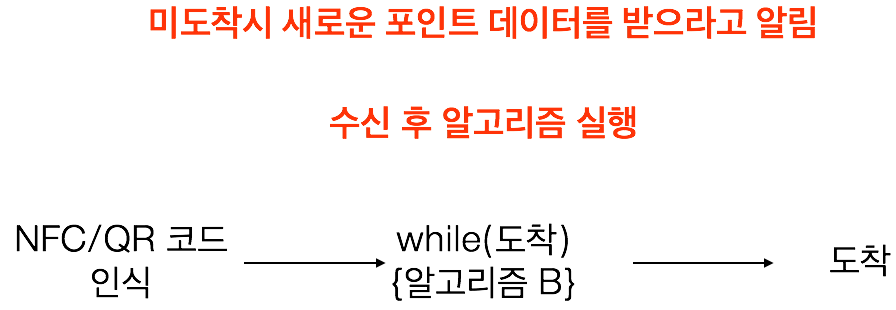
**8. 우리들 만의 장점/차이점**

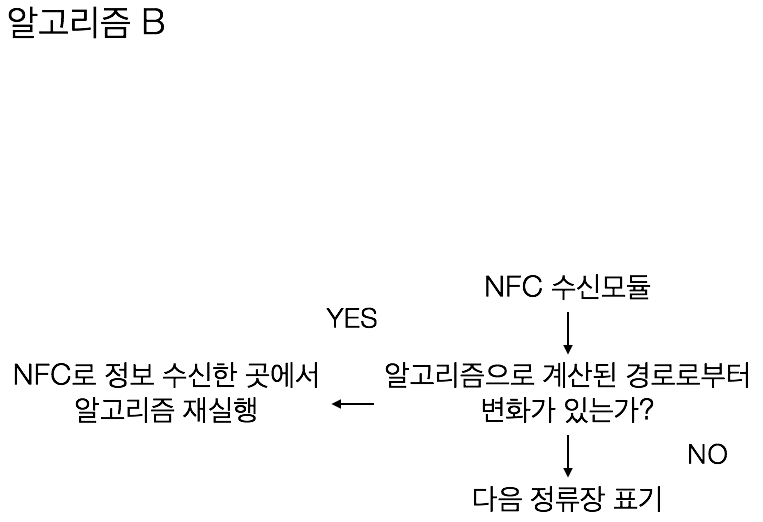
1. 목적지로 가는 도중 현재 자신의 진행도를 확인할 수 있다.
2. 대중교통을 기반으로 하되 보행도 적극 지원한다.
3. 정말 길치를 위한 어플이다.
4. 실시간으로 가장 빠른 루트와 버스를 찾아내기 때문에 매우 효율적이다.

**9. 각 시스템 구성 요소 간의 동작 흐름**









**10. 구성요소 상세 설계**

1) 단말기 : 초기에 지도를 다운로드 받습니다(설치 시). 첫 화면에 나의 위치를 지도에 띄우고 목적지를 입력받도록 합니다.

2) gps모듈 : 위성으로부터 나의 위치를 gps수신기를 통하여 입력 받습니다. 입력 받은 위치를 첫 화면 지도상에 표기합니다. 이 때 지도상에 표기하는 모듈도 구동이 됩니다.

3) 알고리즘A 모듈(?)1. 버스 모듈 : 버스 회사 서버로부터 버스들의 정보를 받는 모듈이 필요합니다. 출발지출부터 가장 가까운 정류장의 위치를 찾아내며 그 정류장에서 목적지로 가는 가장 빠른 버스를 탐색합니다. 그러나 여기서 버스를 기다리는 시간과 운행 시간의 합을 더하여서 시간이 가장 적게 걸리는 코스를 우선적으로 출력해 냅니다.

//목적지로 가는 버스검색

class Search {

string destination;//목적지

string busstation[100];//버스 정거장

public:

void set\_Destination(string D) {

destination = D;

}//목적지 설정

string get\_Destinaiton() {

return destination;

}

void Search\_bus() {

for(int i = 0; i < 100; i++)

{

if(destination == busstation[i])

cout << busstation[i] << endl;//버스가 목적지를 지나가면 출력

}

}

};

//버스 운행 시간(소요 시간)

class Lead\_time {

protected:

int Lead\_time;

public:

void printf() {

cout << Lead\_time << endl;

}//해당 버스의 운행 시간(소요 시간) 출력

};

2. 출력 모듈 : 이 때 출발지로부터 목적지까지 버스의 코스를 추적하여 선으로 이어주는 즉 코스를 출력해주는 모듈이 필요합니다.

//버스 도착 시간 출력

class Arrival : public Lead\_time {

int bustime[100];//버스 도착 시간

public:

void print() {

for(int i = 0; i < 100; i++)

{

cout << bustime[i] << "분" << endl;

}//목적지를 지나가는 모든 버스들의 도착 시간을 출력

}

void print2() {

int arrival\_time = bustime[0];

for(int i = 0; i < 100; i++)

{

if(arrival\_time > bustime[i])

arrival\_time = bustime[i];//가장 빨리 도착하는 버스 시간 찾기

}

cout << arrival\_time << "분" << endl;//가장 빨리 도착하는 버스 시간을 찾아 출력

arrival\_time = bustime[0] + Lead\_time;

for(int i = 0; i < 100; i++)

{

if(arrival\_time > bustime[i] + Lead\_time)

arrival\_time = bustime[i] + Lead\_time;//도착 시간 + 소요 시간이 가장 짧은 버스 시간 찾기

}

cout << arrival\_time << "분" << endl;//도착 시간 + 소요 시간이 가장 짧은 버스 시간 찾아 출력

}

};

4) NFC/QR코드 모듈 : 정류소에 달려있는 NFC송신칩을 통하여 버스 정보를 나의 핸드폰으로 보냅니다. 일반적으로 우리가 사용하는 NFC를 구동하게 하는 모듈이 필요하되 그 기능으로 실시간 버스 정보를 더 정확하게 받을 수 있습니다.

5) 알고리즘B 모듈

1. 알고리즘A와 비교 : 알고리즘 A를 통하여 출력된 코스와 현제 NFC를 찍었을 때 시간상으로 맞다면 그대로 두지만 버스를 놓쳤다거나 더 빨리 도착하여서 다른 버스를 탈 수 있게 되었다면 다시 알고리즘A를 돌립니다.

2. 알람 모듈 : 일정 시간이 지나서 원래 탑승해야 했던 버스를 놓치게 된다면 알람 모듈이 새로운 데이터를 받아 새로운 경로를 출력한다는 알림을 해줍니다.

class NFCmodule : public Search, Arrival{

protected:

int NFC;//새로 받을 NFC

int data;//기존에 받아놓은 NFC

public:

bool NFCdata(int NFC, int data) {

if(NFC == data)

return true;

else

return false;

}//알고리즘으로 계산한 코스에 변화가 있는지 없는지 확인

void app() {

bool boolean;

boolean = NFCdata(NFC, data);

if(boolean == false)//알고리즘으로 계산한 코스에 변화가 있는 경우

{

Search::Search\_bus();//목적지를 지나가는 버스 재출럭

Arrival::print2();//도착 시간 + 소요 시간이 제일 짧은 버스 시간을 찾아 재출력

}

}

};

**10. 시스템 구현에 필요한 기술&구현 환경**

**QR코드 (quick response)**

ISO/IEC 18004

1994년에 덴소웨이브에서 개발되었으나 특허권을 주장하지 않음

기존의 1차원 바코드가 20자 내외의 숫자 정보만 저장할 수 있는 반면,

QR코드는 숫자 7089자, 문자 4296자, 한자 1817자, 이진 2953바이트를 저장할 수 있으며,

일반 바코드보다 인식률과 속도, 복원력이 뛰어남

API나 ZXing(구글에서 제공하는 오픈 소스인 zebra crossing)을 통해서도 만들 수 있음.

애플사의 경우 NFC를 이용할 수 없기 때문에 QR로 대체

**NFC(near fie)**

RFID 기술의 일종으로, 10cm 이하의 거리가 필요하지만 양방향 통신으로 읽기/쓰기를 동시에 진행

복잡한 페어링이 필요 없고 빠른 시간(0.1초) 내에 인식 가능

통신기기 내의 루프 안테나에서 발생하는 자기장을 통해 통신

100-800kbps 의 속도

---

정류장이나 포인트의 위치를 수신하는데 사용. 또한 버스가 도착하는 시간과 다른 차선책에 대해 정보를 전달

**APP 제작 - JAVA**

객체지향프로그래밍 언어로서 C/C++에 비해 간략하고 쉬우며 네트워크 기능의 구현이 용이하기 때문에, 인터넷 환경에서 가장 활발히 사용되는 프로그래밍 언어이다. 자바 프로그램은 운영체제의 종류에 관계없이 대부분의 시스템에서 실행 가능하다.

안드로이드 폰 앱의 개발에 사용되기도 한다.

**GPS(global positioning system)/WIFI를 통한 위치 추적**

지도에 위치를 출력하기 위한 기능

GPS

NAVSTAR 위성들에서 신호를 받음(궤도와 시간)

네트워크 기반 위치(wifi)

접속된 AP의 위치를 기반으로 추적하는 것-민간인이 사용하는 케이스에서 GPS보다는 정확도가 높다.

——————\

A-GPS의 경우 3G,4G 통신망을 이용해 위성 및 시각 정보를 받아오는것-요즘 폰들에 장착

WIFI

초기 지도 데이터베이스를 서버에서 다운로드 받는데 사용

**11. 과제 수행 내용에 대한 각자 의견 및 소감**

이규영 – 벌써 이것을 시작한지 한 학기가 흘러 끝을 맺고 있다. 그냥 느낌으로 봤을 땐 창공설 상위 호환 느낌? 근대 더 상세하게 그리고 더 복잡하며 더 많은 수고가 필요했다. 사실 재미없지는 않았다. 나름 우리조는 즐긴다면 즐긴 조였던 것 같다. 시작은 미비하게 비둘기로 시작하여 끝은 창대하리라. 비둘기로 즐길 것은 다 즐겼고 이후에도 우리만의 언어를 구사했기에 절대 후회는 하지 않는다. 정말 학교와서 보고서 다운 보고서를 쓴 것 같아 뿌듯하다. 비둘기 만세!!!

사실 이걸 표지로 하고 싶었으나 안타깝게도 용지가 세로버전이라 할 수가 없었다.

최지운 - 조별과제에 내가 참여를 잘 안한것 같아 팀원들에게 미안하고 다음부터 조별과제에 더욱 적극적으로 참여를 하여야 겠다고 느꼈고, 직접 씨언어로 알고리즘을 생각해보면서 어플 개발자들이 많은 생각을 하고 힘들게 어플을 개발한다는 것을 알게 되었다.

(솔직히 이미 늦었다.)

임세정 - 비둘기보다 나은 아이디어로 발전되서 정말 기뻤고 조원과 협의하여 창의적인 생각들이 많이 나와서 선택하기가 쉽지는 않았지만 뜻 깊은 시간이었던 것 같다