

핵심만 쏙!

버스전광판

대구소프트웨어고등학교 학생 배수한

2019. 08. 15

1. 서비스 개요

가. 배경

버스의 도착시간, 버스의 현재위치를 알려주는 버스정보시스템은 전국 각지 시내버스 정류장에 보급화 되어있다. 최근 버스정보시스템은 LCD에서 전광판 형태로 바뀌어 날씨, 시간, 뉴스, 홍보영상을 등을 보여주고 있다. 하지만, 대부분의 사람들은 버스 정보 이외 정보는 보지 않는다.

대구소프트웨어고등학교 학생 20명을 대상으로 설문조사를 한 결과 '버스 정보 이외의 다른 데이터를 관심있게 본적 있다'라고 답한 사람이 1명(5%)이었다. 또한, 버스정류장 65세 이상의 노인 10분을 대상으로 한 설문조사에서 '버스 전광판의 글씨가 작아서 뉴스를 읽을 수 없다.'라고 답한 사람이 8명(80%)였다. 또한 한 할머니는 "버스 정보 이외의 다른 데이터 때문에 정신이 사납고 버스 정보만 볼려고 해도 글씨가 작아 잘 안보인다."라고 말했다.

이와같이 버스전광판의 본질을 상실한 채 설치되고 있는 버스정보시스템의 패러다임을 바꾸고 버스전광판의 핵심 기능을 살리기 위해 '핵심만 쏙! 버스전광판'을 개발했다.

나. 서비스 개념

- 버스전광판의 본질을 살린 버스정보시스템
- 노인분들을 위한 큰 글꼴과 정확한 정보
- 시정 홍보를 위한 광고 패널 추가

다. 주요 타겟 고객

1. 전국 시내버스 도로교통과 - 새로운 버스정보시스템을 도입할 고객
2. 전국 시민 (특히 노약자 분) - 버스정보를 필요로 하는 시민

라. 주요 기능

- 1) 전광판 패널
 - 1-1. 인접 버스 도착예정시간 표시
 - 1-2. 잠시후 도착하는 버스 표시
 - 1-3. 시정 홍보 광고 표시
- 2) 시정 홍보 관리 웹사이트
 - 2-1. 시정 홍보 사진 업로드 삭제

2. 서비스 경쟁력/기대효과

가. 경쟁(유사) 서비스 대비 차별화 요소

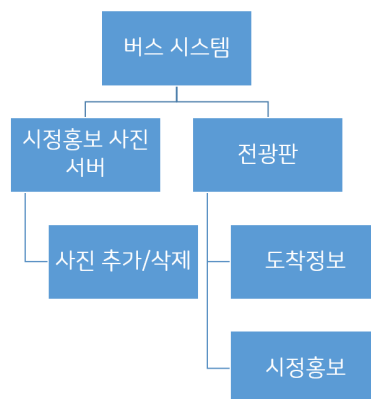
- 1) 본질에 충실한 버스정보시스템
- 2) 큰 글꼴로 노인들이 쉽게 볼 수 있음
- 3) 큼직한 시정 홍보 패널로 시민의 눈길을 끌 수 있음
- 4) 버스 번호 정렬이 규칙적으로 되어있어 자신이 원하는 버스 쉽게 파악 가능
- 5) linux 기반 프로그램으로 보안성이 우수, 저비용으로 보급 가능

나. 기대효과

- 1) 시민과 노인들이 쉽게 자신이 원하는 정보를 찾음
- 2) 더 큼직한 버스 정보 패널로 시정 홍보가 잘됨

3. 프로젝트 계획

가. 프로젝트 구조



나. 프로젝트 일정

개발단계	개발 절차	8월				9월			
설계단계	python, jsp 공부								
	개발 스토리보드 / 디자인								
	서버 구조 설계								
	다이어그램 / 문서화								
제작단계	jsp 서버 개발								
	python gui 틀 완성								
	프로그래밍								
	디버깅 / 테스트								
	사용자 테스트								
	유지보수 및 업데이트								

4. 프로젝트 결과

가. 프로젝트 계획서에 적힌 모든 내용을 충족하여 개발 완료 되었음

이미지 서버



라즈베리파이 원격접속 한 모습(전광판)

노선번호	도착예정	현재위치
300	13분	판문동현대아파트(18전)
350	3분	봉곡동시외정류소(4전)
352	15분	내동초교(17전)
370	3분	봉곡동시외정류소(4전)
390	6분	이현상가(9전)

잠시후
134 253 350 370

gui 코드

```
import sys
from time import sleep
import threading
import re
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
import urllib.request
from urllib import parse
import requests
import xml.etree.ElementTree as ET
import math
from Ui_design import *
```

```
img_sleep = 15
data_sleep = 5
jamsiwho = 180
```

```
def img_thread():
```

```
    while(True):
```

```
        img_text = requests.get(
```

```
"http://10.80.161.248:8080/Image_server/getlink.jsp").text.replace("\r",
 "").replace("\n", "")
```

```
        img_list = img_text.split("|")
```

```
        for i in range(0, (len(img_list)-1)):
```

```
            img_get(img_list[i])
```

```
            sleep(img_sleep)
```

```

def img_get(url):
    try:

        image = urllib.request.urlopen(url).read()
        pixmap = QtGui.QPixmap()
        pixmap.loadFromData(image)
        ui.imgPanel.setPixmap(pixmap)
        ui.imgPanel.setScaledContents(True)
    except:
        print("이미지 없네")

```

```

def get_location(nodeid, routeid, arrprev):

```

```

    url = urllib.parse.quote(

"GnI6jLB1CE+8FW+gJ+JiXHGMbvYR0TSHxkKdJmVCX4z1/YDusIIlgC5FjkUTf+I7dcj
u568zwerhryR5NaEzg==")

    data =
requests.get("http://openapi.tago.go.kr/openapi/service/BusRouteInfolnquireServic
e/getRouteAcctoThrhgSttnList?serviceKey=" +

url+"&pageNo=1&numOfRows=100000&cityCode=38030&routeId="+routeid).text
    root = ET.fromstring(data)

    nodeord = -1

    for item in root.iter('item'):
        if(str(item.find('nodeid').text) == str(nodeid)):

            nodeord = int(item.find('nodeord').text)
            break

```

```

for item in root.iter('item'):
    if(int(item.find('nodeord').text) == nodeord-arrprev):

        return str(item.find('nodenm').text) + "("+str(arrprev)+"전)"

def data_thread():
    while(True):
        data = requests.get(

"http://openapi.tago.go.kr/openapi/service/ArvlInfoInquireService/getSttnAcctoArvl
PrearngelInfoList?serviceKey=Gnl6jLB1CE%2B8FW%2BgJ%2BjiXHGMbvYR0TSHxk
KdJmVCX4z1%2FYDuslllgC5FjkUTf%2BI7dcju568zwerhryR5NaEzg%3D%3D&cityC
ode=38030&nodeId=JJB381227002&numOfRows=100000").text

        root = ET.fromstring(data)

        cnt = int(root.find('body').find('totalCount').text)
        na = math.ceil(cnt/6)

        if(na == 0):
            sleep(data_sleep)
            next

        cnt_list = []
        for i in range(na):
            cnt_list.append(cnt//na)
            if(i < cnt % (cnt//na)):
                cnt_list[i] += 1

        items = root.iter('item')

# =====
i = 0

```

```

j = 0

no_list = []
time_list = []
nolo_list = []

now_str = ""

for item in root.iter('item'):
    t = int(item.find('arrtime').text)

    if(t <= jamsiwho):
        now_str += str(item.find('routeno').text) + "  "

ui.now_num.setText(now_str)

for item in items:
    i += 1

    no_list.append(item.find('routeno').text)
    t = int(item.find('arrtime').text)
    time_list.append(str(round(t/60))+"분")

    nolo_list.append(get_location(item.find('nodeid').text,
                                item.find('routeid').text,
                                int(item.find('arrprevstationcnt').text)))

    if i == cnt_list[j]:
        ui.bus_num.setText(
            '<html><head/><body><p                                style="line-height:
1.15"><span>'+<br>'.join(no_list)+'</span></p></body></html>')
        ui.bus_time.setText(
            '<html><head/><body><p                                style="line-height:
1.15"><span>'+<br>'.join(time_list)+'</span></p></body></html>')

```



```
        ui.bus_nolo.setText(  
            '<html><head/><body><p'                                style="line-height:  
2.57"><span>'+'<br>'.join(nolo_list)+'</span></p></body></html>')
```

```
        i = 0  
        j += 1
```

```
        no_list = []  
        time_list = []  
        nolo_list = []
```

```
        sleep(data_sleep)
```

```
app = QtWidgets.QApplication(sys.argv)  
MainWindow = QtWidgets.QMainWindow()  
MainWindow.showFullScreen()  
# global ui  
ui = Ui_MainWindow()  
ui.setupUi(MainWindow)
```

```
MainWindow.setCursor(QtCore.Qt.BlankCursor)
```

```
thread = threading.Thread(target=img_thread, args=()) # 쓰레드 생성  
thread.daemon = True  
thread.start()
```

```
thread = threading.Thread(target=data_thread, args=()) # 쓰레드 생성  
thread.daemon = True  
thread.start()
```

app.exec_()