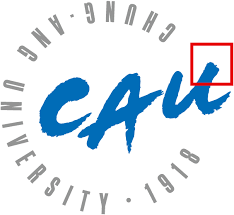
Data Structure Design(02)\_ Testing & Evaluation

Individual Project – 유동인구와 미세먼지 그리고 관심사를

이용한 스마트 데이트 코스 추천 Application



|  |  |
| --- | --- |
| Department | Computer Science Engineering |
| Name(Student Number) | Kwon Dokyeong (20154077) |

# Testing Case Select

## Testing Case Select

본 프로젝트의 최종 결과를 테스팅 하기 위한 목록으로 크게 3가지를 설정하였다. 첫 번째로 Application이용시에 발생할 수 있는 시스템 적인 에러가 잘 해결이 되어 있는지, 두 번째로 Application에서 제공하는 루트가 주어진 자료에서 최적의 선택이 맞는지에 대한 테스팅 그리고 마지막으로 여러 상황에 맞추어 알맞은 결과를 내보내는지에 대한 테스트를 진행하려고 한다.

## Testing Case Table

|  |  |
| --- | --- |
| System Error Testing | Null이 입력되었을 때 오류 처리 테스팅 |
| 잘못된 정보가 입력되었을 때 오류 처리 테스팅 |
| 선택 장소에 모든 가게가 영업하지 않을 때에 관한 테스팅 |
| 사용자 입력 정보의 모순 테스팅 |
| 미세먼지 정보 오류 처리 테스팅 |
| 유동인구 정보 업데이트 테스팅 |
| Algorithm Testing | 구 선택한 알고리즘의 타당성 테스팅 |
| 데이트 장소 선택 알고리즘의 타당성 테스팅 |
| Other Testing | 구 선택 가중치 출력 테스팅 |
| 장소 선택 가중치 출력 테스팅 |
| 우선순위 출력 테스팅 |

# Testing Result

## System Error Testing Result

1. Null값 Error Testing

사용자가 본 Application을 사용하면서 입력하는 정보는 2가지가 있다. 첫 번째는 현재 위치이고, 두 번째는 선호하는 데이트와 데이트 관련 정보이다. 이 두가지에 대한 Null Value Testing 결과는 아래와 같다.

* 현 위치 Null 값 입력

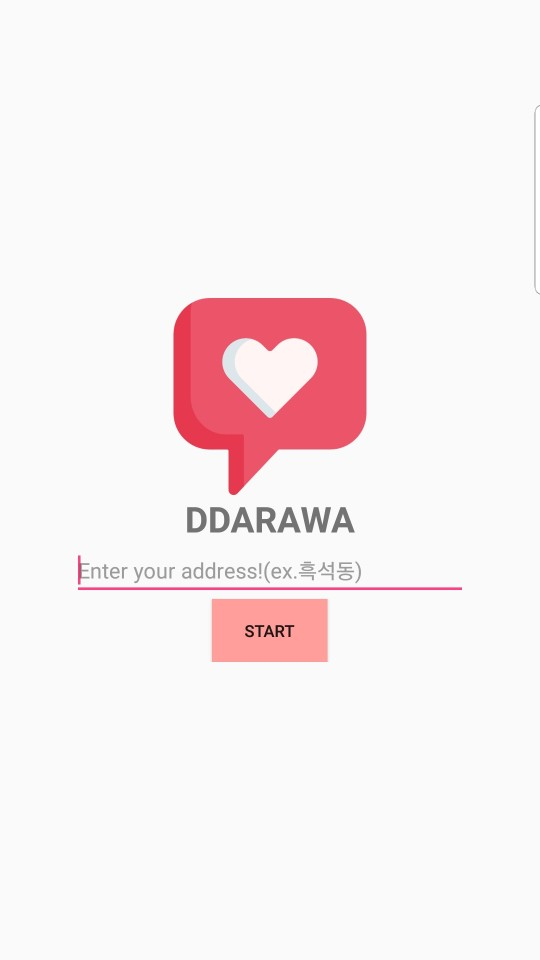


Figure 1

아무것도 입력하지 않은 상태에서 Start를 누르게 되면 다음 Activity로 넘어갈 수 없게 오류를 잘 해결하였다.

* 데이트 정보 Null 입력

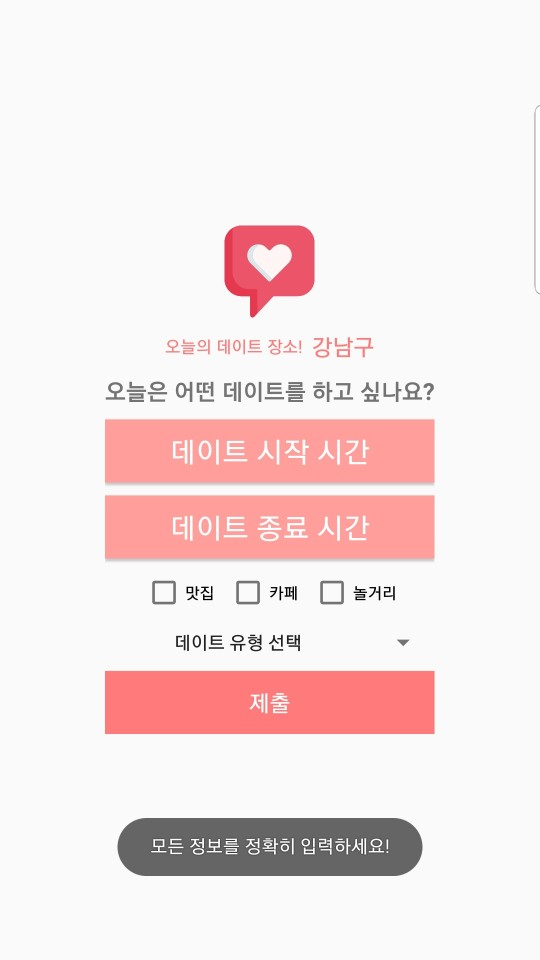


Figure 2

정보 누락 시 Toast 메시지를 이용해 ‘모든 정보를 정확히 입력하세요!’라는 문구를 띄워 시스템 오류를 잘 처리함을 확인하였다.

1. 잘못된 정보 입력 Testing

본 Application에서 잘못된 정보를 입력할 수 있는 상황은 현재 위치를 선택하는 상황이다. 이 상황에 대한 Testing 결과는 아래와 같다.

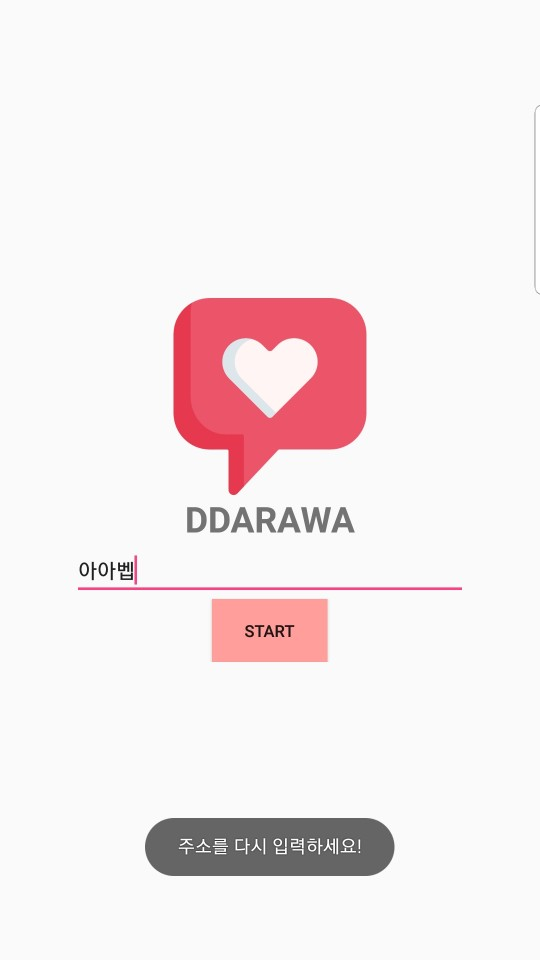


Figure 3

Figure 3과 같이 Toast message를 띄워 ‘주소를 다시 입력하세요!’라는 알림을 보내 사용자가 입력한 위치 정보가 잘못되었음을 알려줄 수 있다.

1. 영업시간 외 시간 입력 Testing

본 Application에는 데이트 시간을 입력하는 상황이 있다. 현재 선택 된 위치의 데이트 장소들의 영업시간이 사용자가 입력한 데이트 시간과 맞지 않을 때의 오류 처리 결과는 아래와 같다.

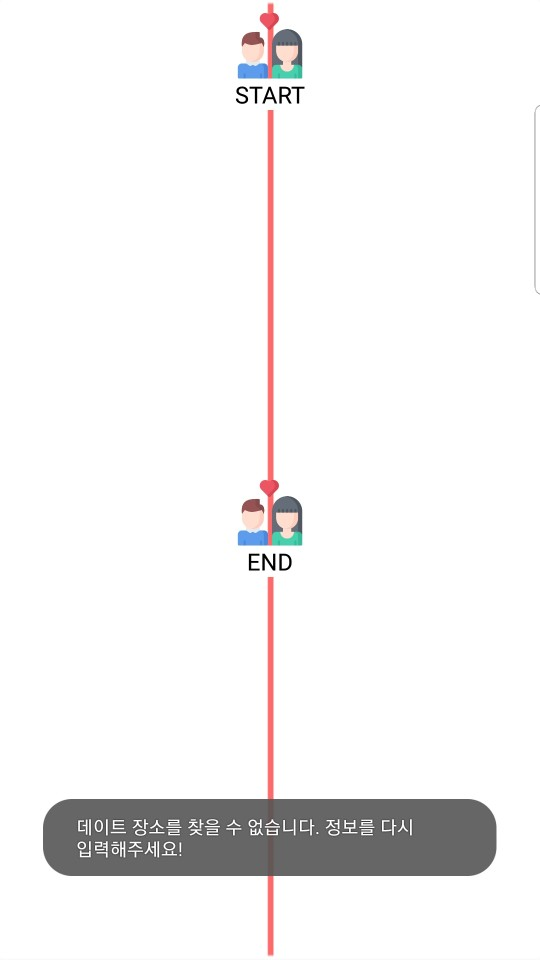
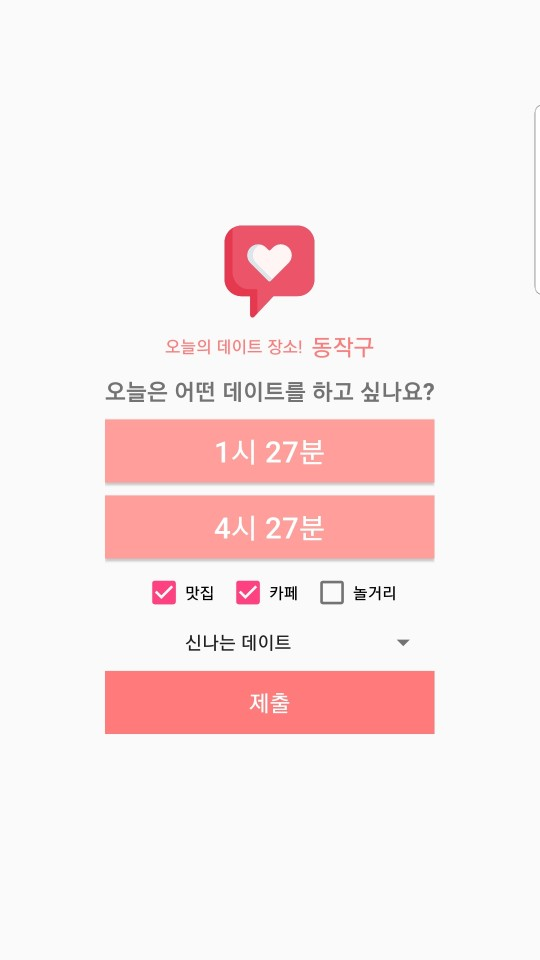


Figure 4

Figure 4와 같이 입력 시간이 새벽 1시~4시 인 경우, 동작구의 24시간 영업을 하는 데이트 장소가 없다면 Figure4의 오른쪽과 같이 Toast 메시지를 띄워 사용자가 입력을 다시 받을 수 있도록 요구한다.

1. 입력 정보 오류 Testing

사용자가 입력하는 정보에서 오류가 날 수 있는 부분은 한가지이다. 예를 들어 맛집, 카페, 놀거리 3가지의 카테고리를 선택을 하였는데, 데이트 총 시간은 1시간도 안되는 경우이다. 이런 경우는 발생할 수 없기 때문에 미리 처리를 하여 데이트 정보 입력이 조금 더 정확해지도록 요구하였다. 처리 결과는 아래와 같다.

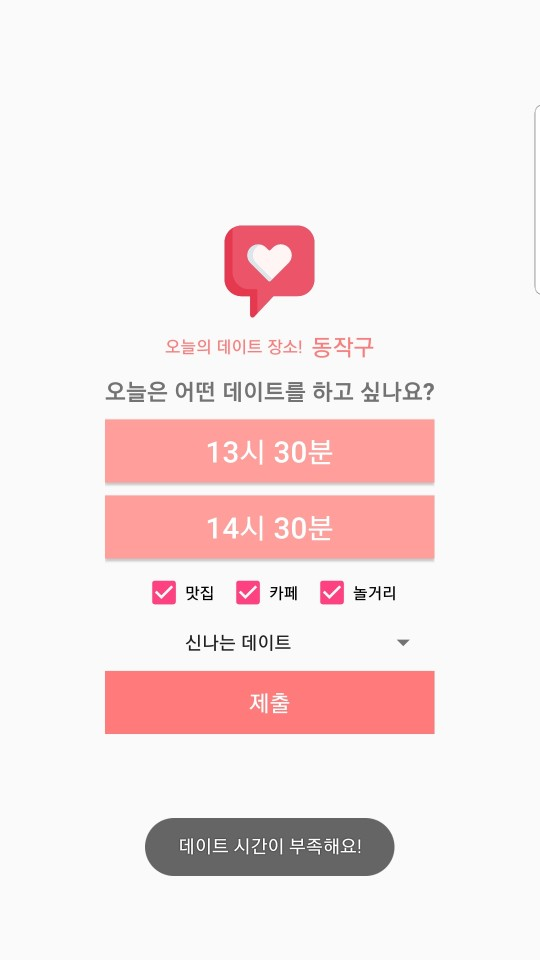
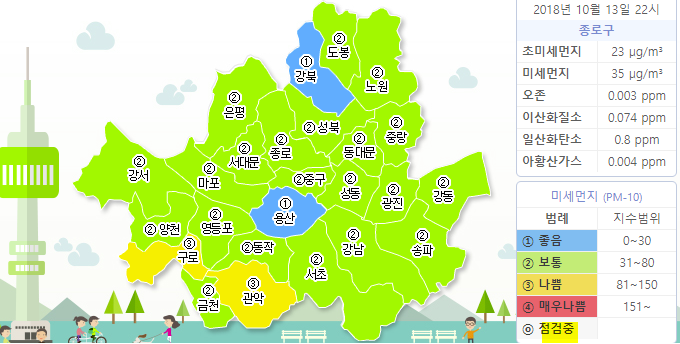
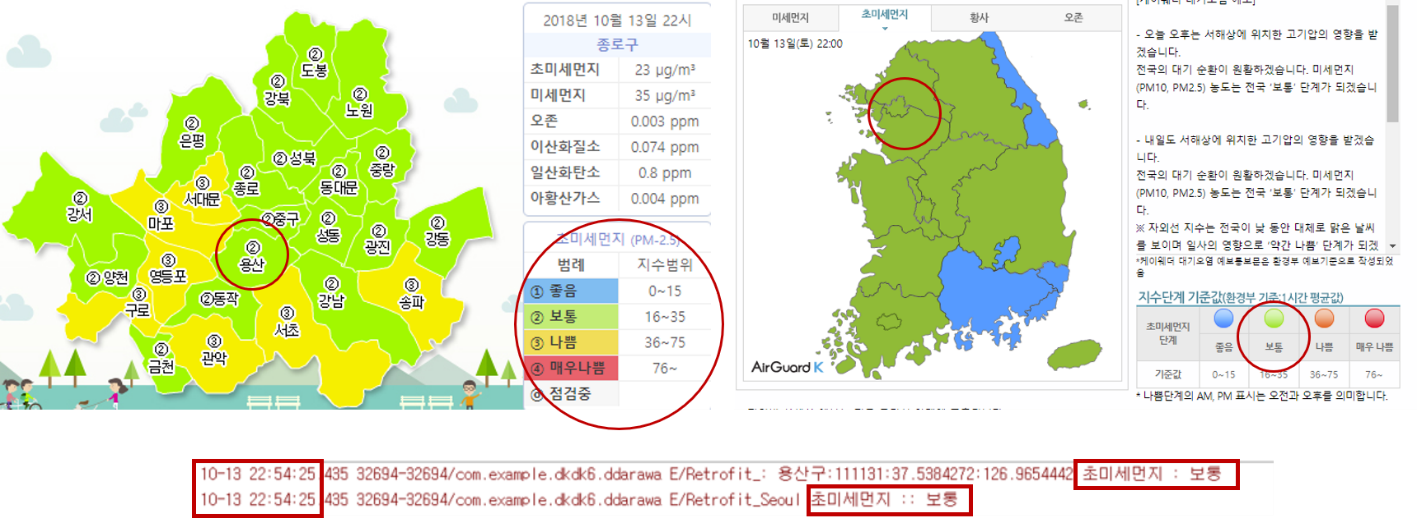


Figure 5

Figure5와 같이 Toast메시지를 띄워 사용자의 입력을 다시 받을 수 있도록 하였다. 또한 이런 오류 처리에 대한 시간 측정은 한 카테고리당 약 1시간정도의 데이트를 진행하여야 한다는 가정을 전제로 처리 코드를 작성하였다.

1. 미세먼지 정보 오류 처리 Testing





Figure

서울시 공공 데이터를 활용하고 있는 만큼, 서울시에서 올바른 데이터를 받아 와야 본 Application도 정상 작동이 될 수 있다. 여기서 문제점은 서울시 모든 행정구역에 있는 미세먼지 측정소 중 가끔 ‘점검중’이라는 상황이 발생한다는 것인데(Figure 6의 오른쪽 하단 형광팬 부분 참조), 이런 상황이 발생했을 때 본 어플리케이션은 서울시에서 제공하는 전체적 미세먼지 평균치를 이용하여 처리하였다. Log를 찍어본 결과는 Figure6의 아랫부분에 있는 그림과 같이, 해당 행정구역의 미세먼지 정보와 서울시의 평균 미세먼지 정보를 보여준다. 이 사진은 2018.10.13일 동일한 시간대에 각 행정구별 초미세먼지 측정 결과와 서울시의 평균 초미세먼지 정보, 그리고 직접 안드로이드 스튜디오에서 Log를 찍어 살펴 본 결과가 모두 동일하게 오류없이 출력함을 보여준다.

1. 유동인구 정보 업데이트 Testing

서울시 공공 데이터에서 유동인구 정보는 1달에 1번씩 업데이트 됨을 확인하였다. 그래서 본Application은 매달 1일을 중심으로 정보를 갱신해올 수 있도록 http통신 쿼리를 작성하였고, Postman을 이용하여 쿼리문을 Testing하였다. 처리 관련 코드와 테스팅 결과는 아래와 같다.

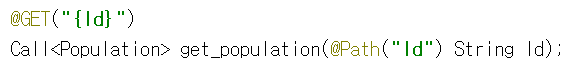


Figure 7

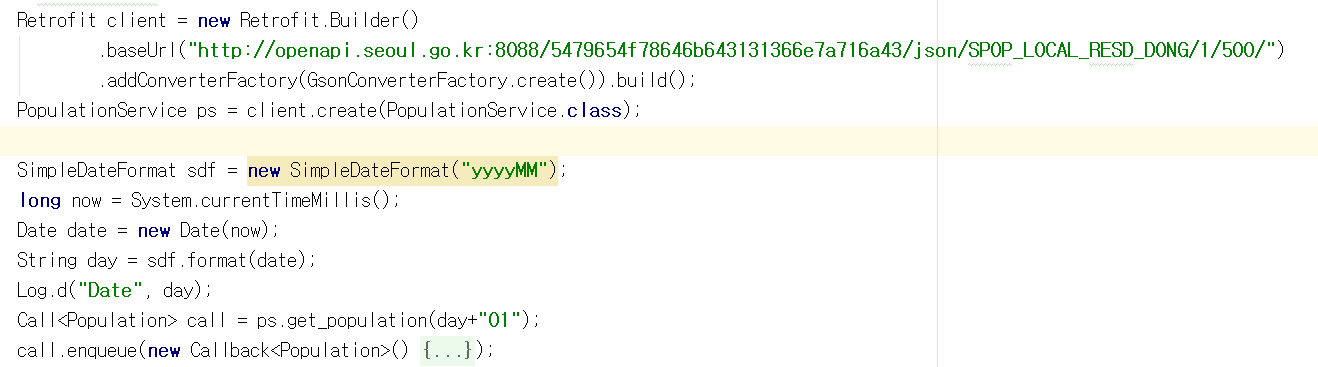


Figure 8



Figure 9

Figure8과 Figure9에서 볼 수 있는 것처럼, 현재 일자에 맞는 월수를 구해 매달 1일마다 유동인구 데이터를 갱신할 수 있도록 구현하였고, Figure 9의 안드로이드 Log 결과와 같이 테스팅 시에도 오류 없이 데이터 갱신 일자가 출력됨을 확인하였다. Figure 9에서의 왼쪽 붉은 박스는 테스팅 시점 날짜이고 오른쪽의 붉은 박스는 데이터 갱신 일자를 표현하고 있다.

## Algorithm Testing Result

1. 행정구역 선택 Testing

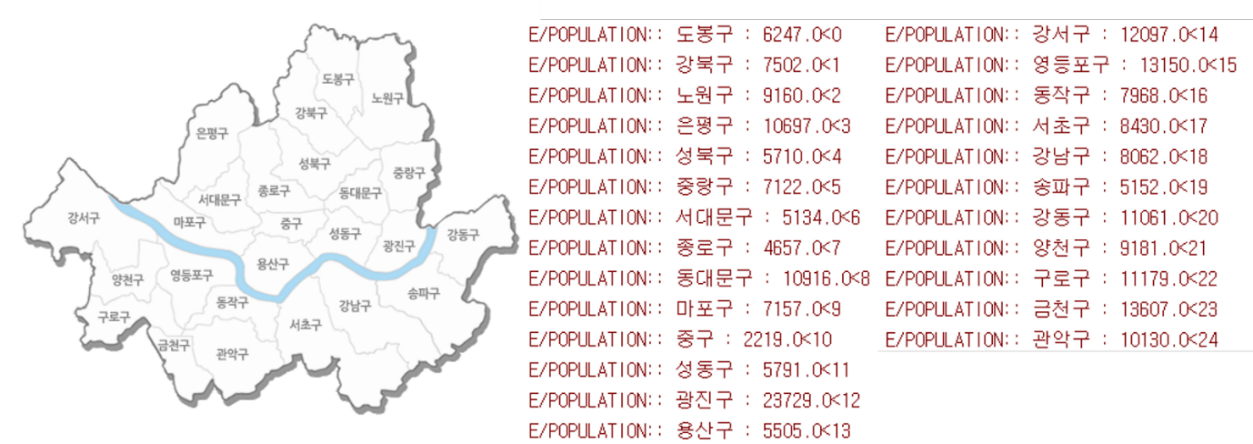


Figure 10

우선 Figure 10는 현재 서울시의 지도를 행정구역별로 나타낸 그림과 현재 서울시의 20~30대 인구수를 계산한 결과를 출력한 것이다. 이 정보를 바탕으로 사람이 가장 많은 광진구와 사람이 가장 적은 송파구 그리고 평균 정도의 인구가 있는 노원구에 대한 테스트를 진행해보았다.

* 송파구 Testing



Figure 11

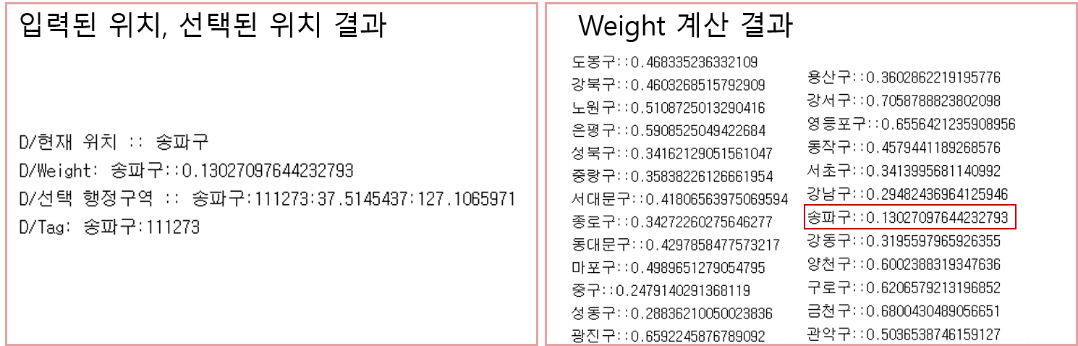


Figure 12

Figure 10의 인구 정보를 바탕으로 Figure11에서는 송파구 주변 6개 구역을 계산된 서울 평균 유동인구를 바탕으로 3단계로 나누어 표현하였다. 송파구는 약 5000명으로 현재 주변 구역 중 가장 낮은 인원수를 보이고 있으므로 송파구를 검색했을 때 예상하는 결과 역시 송파구이다. 테스팅 결과는 Figure 12로, 예상과 동일한 결과를 보였다.

* 노원구 Testing

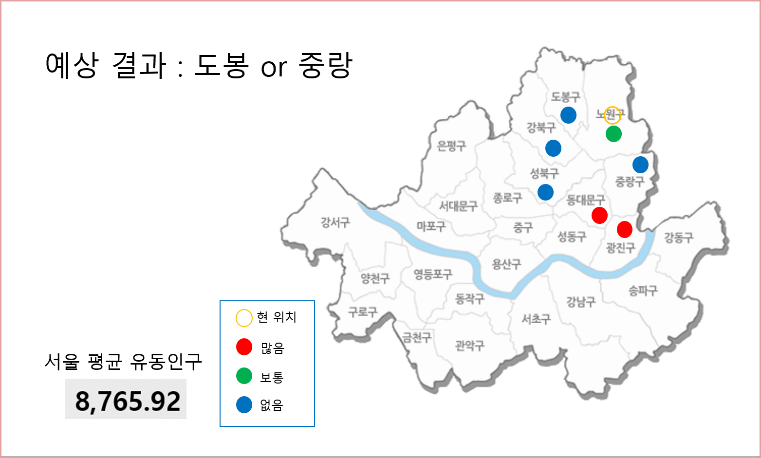


Figure 13

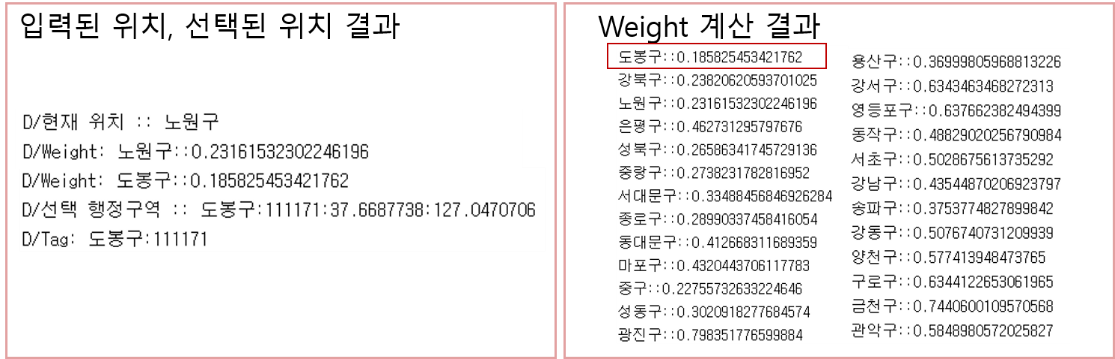


Figure 14

Figure 10의 인구 정보를 바탕으로 Figure13에서는 노원구 주변 6개 구역을 계산된 서울 평균 유동인구를 바탕으로 3단계로 나누어 표현하였다. 노원구는 약 9000명으로 현재 주변 구역 중 가장 평균적인 인원수를 보이고 있는데, 주변의 행정구역 대부분의 유동인구 정보가 ‘없음’이라는 것을 확인할 수 있다. 이 정보와 거리적 정보를 바탕으로 생각하였을 때 도봉구 혹은 중랑구가 선택이 될 것이라고 예상을 하였고, 테스팅 결과는 Figure 14으로, 예상과 동일한 결과를 보였다.

* 광진구 Testing

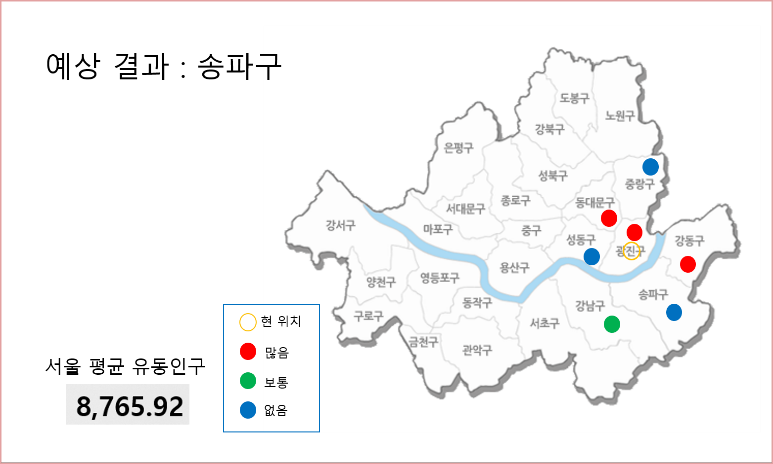


Figure 15

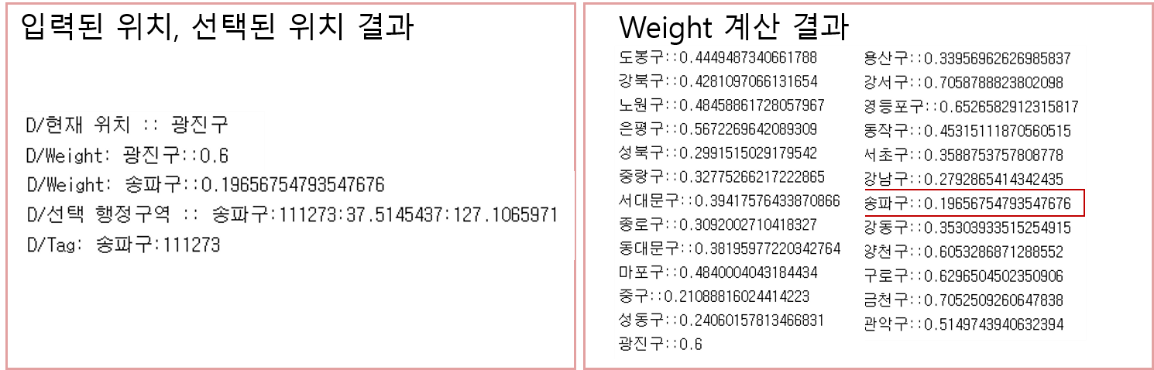


Figure 16

Figure 10의 인구 정보를 바탕으로 Figure15에서는 광진구 주변 6개 구역을 계산된 서울 평균 유동인구를 바탕으로 3단계로 나누어 표현하였다. 광진구는 약 23000명으로 현재 주변 구역 중 가장 많은 인원수를 보이고 있는데, 주변의 행정구역도 다양한 인구 분포를 보이고 있었다. 광진구에서 선택이 가능 한 구역은 성동구 or 송파구 일텐데, 송파구의 인원은 서울시에서 가장 적은 약 5000이었다. 이 정보와 거리적 정보를 바탕으로 생각하였을 때 거리는 송파구와 성동구가 비슷하지만 송파구의 인원이 훨씬 적으므로 송파구가 선택이 될 것이라고 예상을 하였고, 테스팅 결과는 Figure 16로, 예상과 동일한 결과를 보였다.

1. 데이트 장소 선택 Testing

장소 선택에 대한 Testing 결과를 보이기 위해, 위에서 언급하지 않았던 행정구역인 동작구를 바탕으로 Testing 결과를 보이려고 한다. 데이트 장소 선택의 가중치 바탕 예상 결과와 실제 결과를 비교해보면 아래와 같다. 우선 동작구의 현재 입력되어 있는 데이트 장소를 마크로 찍어보면 Figure 17과 같다.

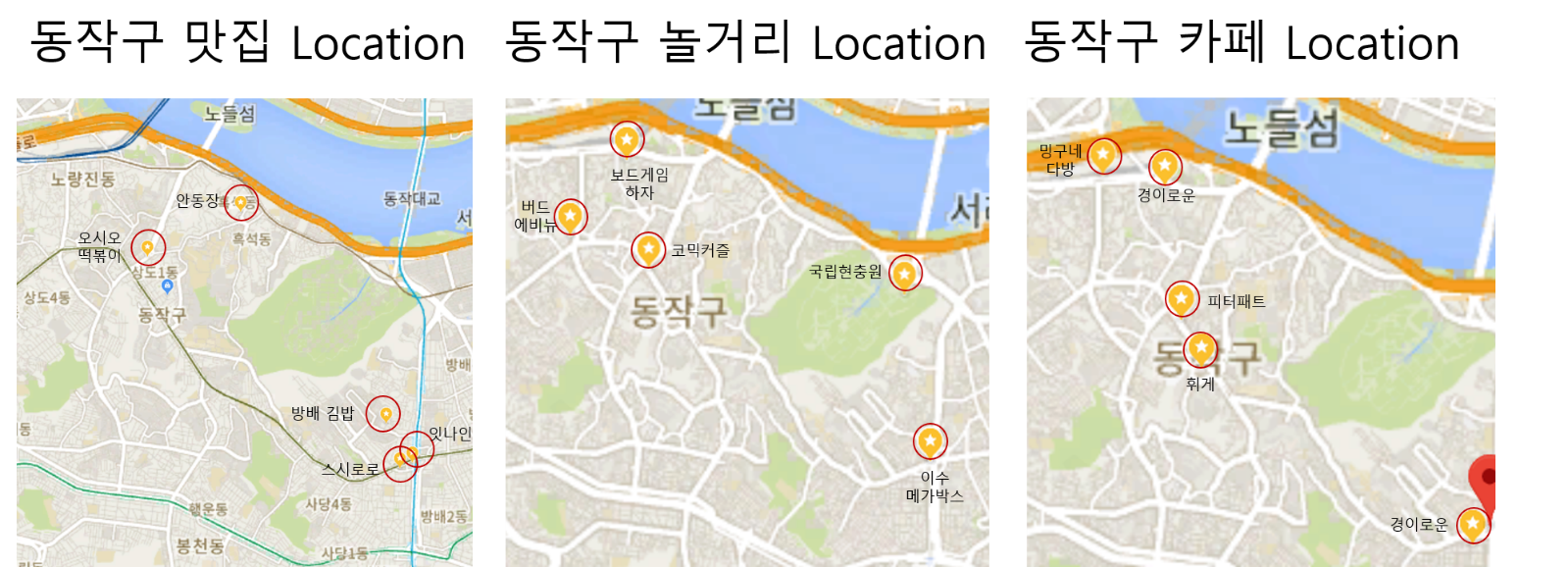
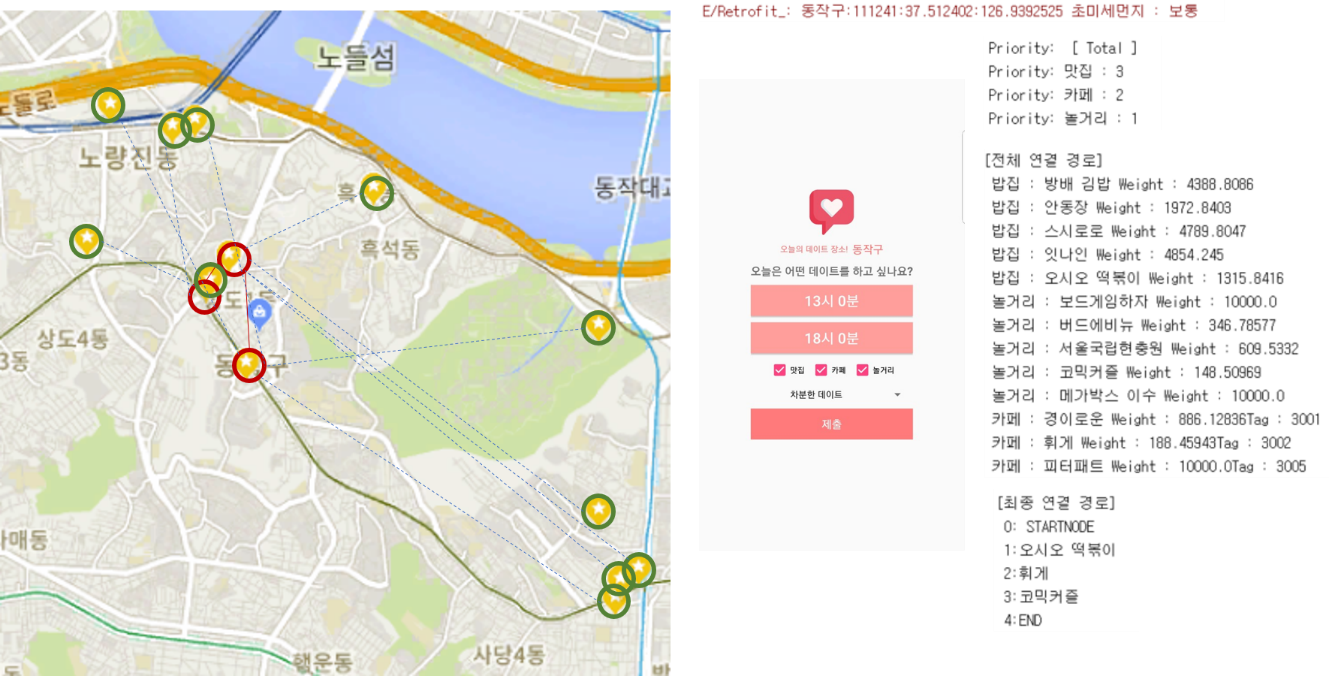


Figure 17

이 데이트 장소 정보를 바탕으로, 동작구에서 ‘차분한 데이트’를 선택하였을 때와 그리고 ‘신나는 데이트’를 선택하였을 때 어떤 결과가 출력되는지, 가중치는 어떠 한지 출력해 본 결과는 아래와 같다.

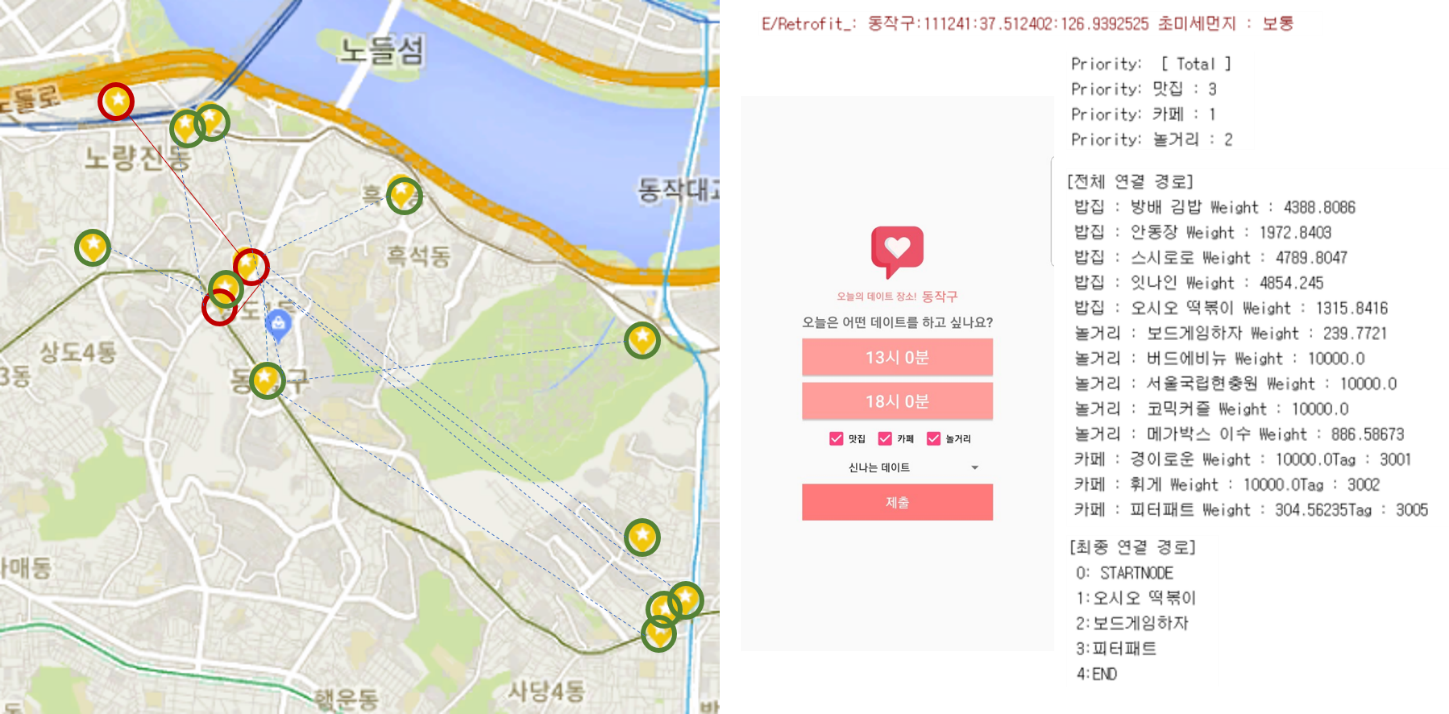
* 동작구, 차분한 데이트 선택



Figure

동작구의 데이트 장소는 Figure 18의 지도와 같은 부분인데, 현재 동작구 초미세먼지 상태가 보통이므로 데이트 장소간 거리는 그렇게 많이 떨어지지 않은 곳으로 추천하면서, 차분한 데이트이므로 ‘조용한’과 같은 Tag가 붙은 장소를 추천해주어야 한다는 것이 예상하는 결과였고, Figure 18의 오른쪽처럼 코믹커즐은 북카페이고 휘게도 조용하다고 알려진 카페이고, 시작 노드인 오시오떡볶이로부터 가까운 거리에 있는 장소들을 보여준다는 것을 지도의 빨간색 노드들을 확인해보면 알 수 있다.

* 동작구, 신나는 데이트 선택

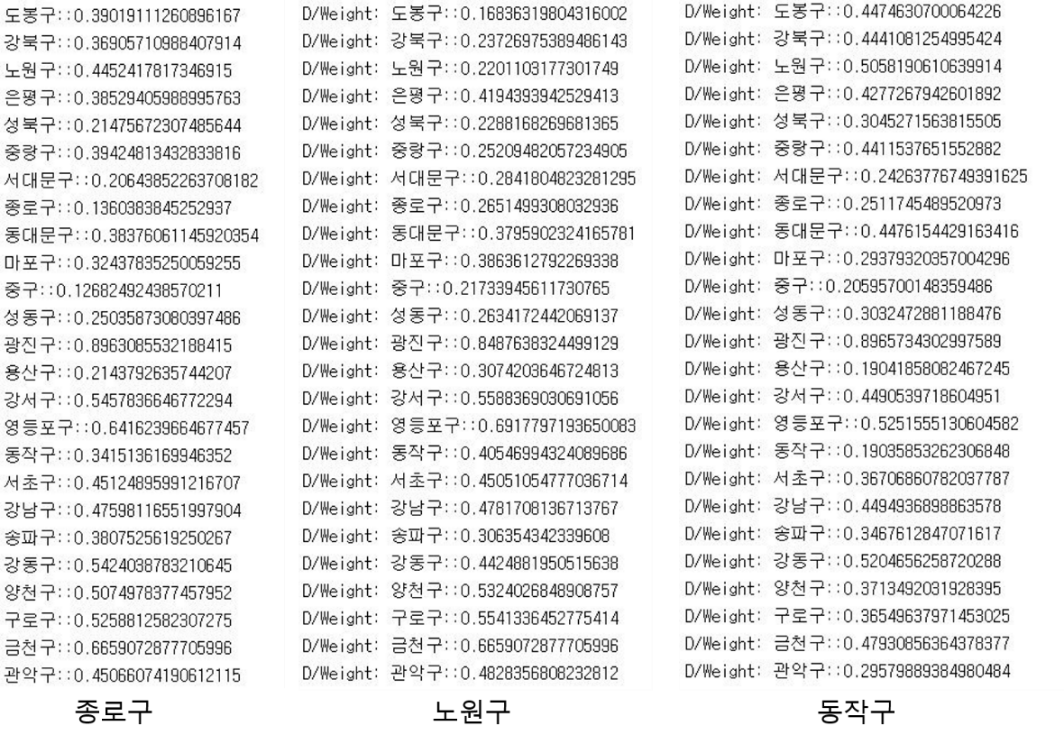


동작구의 데이트 장소는 Figure 19의 지도와 같은 부분인데, 현재 동작구 초미세먼지 상태가 보통이므로 데이트 장소간 거리는 그렇게 많이 떨어지지 않은 곳으로 추천하면서, 신나는 데이트이므로 ‘활동적인, 신나는’과 같은 Tag가 붙은 장소를 추천해주어야 한다는 것이 예상하는 결과였고, Figure 19의 오른쪽처럼 보드게임하자는 보드게임을 할 수 있는 활동적인 카페이고 피터패트는 사진찍기 좋기로 알려진 카페이고, 시작 노드인 오시오떡볶이로부터 가까운 거리에 있는 장소들을 보여준다는 것을 지도의 빨간색 노드들을 확인해보면 알 수 있다.

## Other Testing Result

1. 행정구역 가중치 출력 Testing

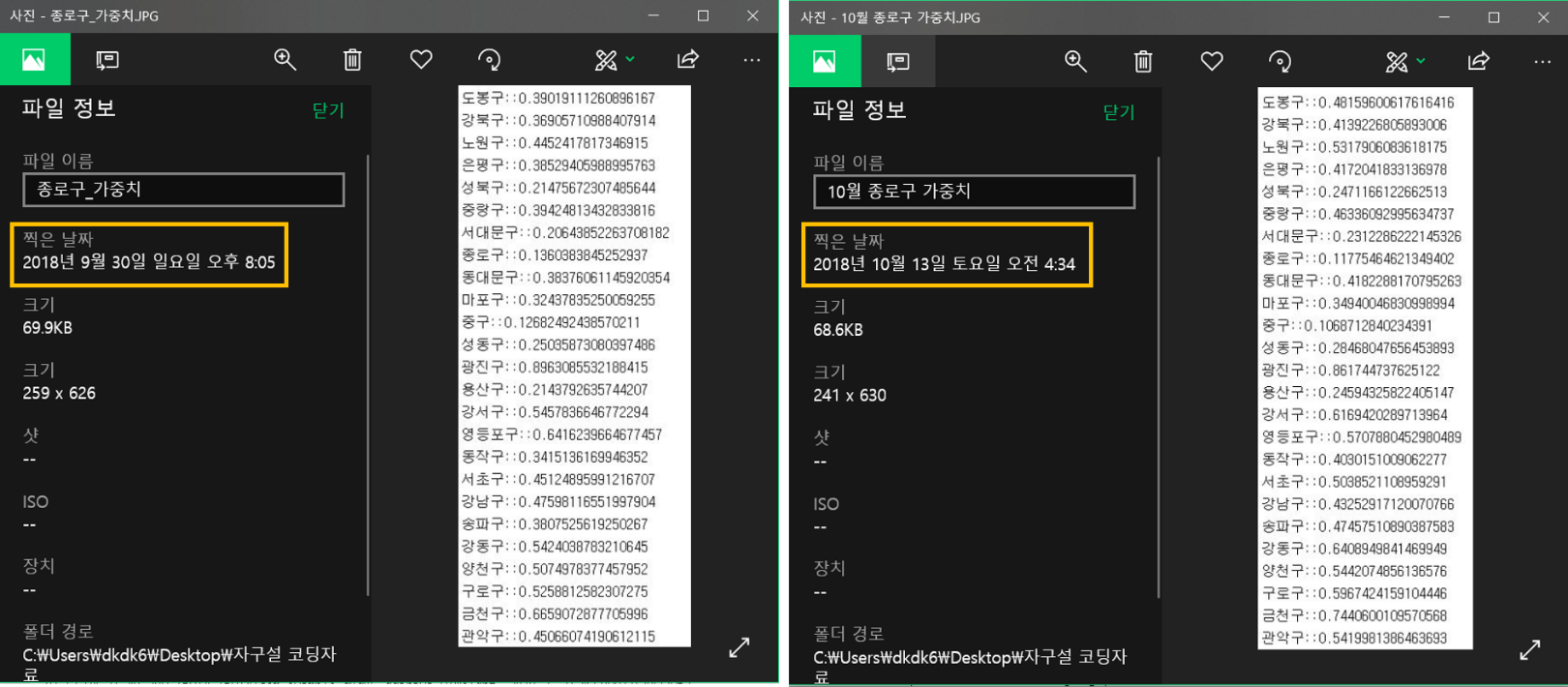
* 행정구역 가중치 출력 확인



Figure

Figure19와 같이 다양한 구를 선택시에도 오류 없이 각자 다른 Weight을 계산하는 것을 확인할 수 있었다.

* 데이터 갱신 후 가중치의 변화 확인



Figure

Figure20의 노란 박스에서 확인할 수 있는 것처럼 1달마다 정보를 갱신하기 때문에 같은 행정구역인 ‘종로구’였지만 9월과 10월의 가중치가 잘 변화하였음을 확인하였다.

1. 장소 선택 가중치 출력 Testing



Figure

Figure21과 같이 다양한 상황에서도 오류 없이 각 장소에 대한 Weight을 계산하는 것을 확인할 수 있었다.

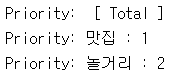
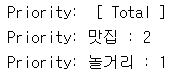
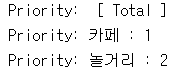
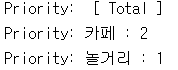
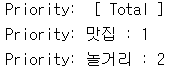
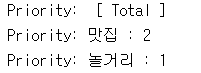
1. 우선순위 출력 Testing

본 Application에서는 첫 Node 선정에 있어 우선순위를 주어 매번 다르게 경로를 출력하고자 하였고, 각 케이스별 우선순위가 정상 출력되는지 확인하였다. 결과는 아래와 같이 한쪽에 치우치지 않고 상황별로 다양한 결과를 출력하는 것을 확인할 수 있었다.

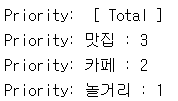
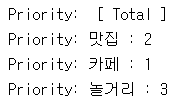
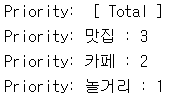
* 카테고리 1개 선택 시



* 카테고리 2개 선택 시]



* 카테고리 3개 선택 시



# Evaluation

## Dynamic 요소

* Node

우선 Node에서 Dynamic요소를 찾아보면 행정구역 즉, City 선택 시는 Node는 변화 별로 없다. 하지만 데이트 장소에 대한 Node 정보는 사용자의 데이트 입력시간과 각 가게의 영업시간을 고려하여 없어지는 Node와 생겨나는 Node가 생긴다는 점에서 Dynamic하다고 할 수 있다.

* Edge

Edge에 대한 Dynamic요소는 City를 선택하는 상황과 Place를 선택하는 두 상황 모두 Weight이 실시간 입력 받은 정보에 따라 Weight이 실시간으로 변하기 때문에 Dynamic하다고 할 수 있다. 우선 행정구역 선택 시에는 유동인구 정보에 따라 계속 업데이트되고 바뀌는 Edge의 Weight이 있고, 데이트 장소 즉 가게 선택시에는 Weight이 미세먼지 정보에 따라 가중치가 바뀌므로 Dynamic하다.

## Graph 요소

이 프로젝트는 루트 추천이 최종적 목표이다. 또한 선택된 장소로부터 연결 되어있는 모든 장소 중 가장 최단을 찾아내고 있기 때문에 Graph로 반드시 표현해야 하며, 최종적 결과를 출력하는 방법 역시 Graph로 해결이 가능하기 때문에 Graph적 요소도 잘 충족시키고 있다고 할 수 있다.