|  |
| --- |
| **1. 주제**  이동 시간 효율화를 위한 GPS Tracking 기능 개발  **분반, 팀, 학번, 이름**  가반, 10팀, 20231762, 문재민 |

|  |  |
| --- | --- |
| **2. 요약**  요일이나 시간에 따라 소요 시간이 크게 달라지는 길들이 존재한다. 자신만의 DB를 만들어 이동시간을 기록해 시간을 더 효율적으로 쓰도록 도울 수 있는 기능을 만들어보려 한다.  Python의 folium 라이브러리는 지도를 다룰 수 있도록 도와준다. time 라이브러리는 시간을 다룰 수 있도록 도와준다. 이동경로를 따라 지도 위에 자취를 남기고 이동시간을 기록한다. 중간에 들린 곳이 있다면 그 시간은 제외하고 측정하는 로직도 코딩한다. 이동 경로와 소요 시간은 DB에 저장한다. 필요하다면 추가적으로 comment를 기록하는 기능도 추가한다. 이 데이터들은 pycopg2 라이브러리를 이용해 DB에 저장한다. 저장된 데이터는 필요에 따라 쉽게 열람할 수 있도록 한다.  서울 같이 시간에 따라, 요일에 따라 도로 상황이 크게 바뀌는 도시에서는 네비게이션도 큰 역할을 하지 못한다. 그럼에도 택시기사님들은 상황에 따라 가장 빠른 길을 경험에 의해 알고 계신다. 비교적 적은 노력으로 사용자도 그런 경지에 도달할 수 있도록 도와준다. 그리고 운전 이외에도 자주 가는 산책길이나 걸어서 다녀야 하는 길들도 기록할 수 있다. 하나의 기록이 끝나면 기록할 수 있는 comment를 이용해 산책 일기장 같은 용도로도 사용할 수 있다. | **3. 대표 그림**  개발배경: 운전을 하고 다니며 교통체증에 스트레스를 많이 받았다. 같은 길이라도 시간이나 요일에 따라 도로 상황이 천차만별이었다. 이런 상황에서 조금이라도 더 소요시간을 줄이고, 운전이라는 행위에 성취감을 더하고 싶었다.  그림1) GPS Tracking UI  예상 결과: 단기간에 큰 변화를 불러올 것이라고는 생각할 수 없지만, 꾸준히 한다면 웬만한 사람들보다 지리에 빠삭한 사람이 될 수 있다. 이외에도 그저 산책이나 조깅 같은 취미 생활에도 관성이 작용하도록 도와줄 수 있을 것이라 기대한다. |

|  |
| --- |
| **4. 서론**  서울과 같이 교통량이 많은 도시는 시간이나 요일에 따라 교통 상황이 크게 달라진다. 같은 길이라도 어느 요일에, 언제 나가냐에 따라 소요 시간이 천차만별이다. 그 증거는 <서울교통 빅데이터 플랫폼>에 접속하면 쉽게 찾을 수 있다. 서울 교통 현황 중 속도에 관한 부분을 보면 ‘원활’ 상태를 찾기 힘들다. 가장 많은 상태는 ‘서행’ 단계인데, 평균 속도가 15~25km/h임을 나타낸다. 그마저도 통행량이 많아지는 출/퇴근 시간이나 점심 시간에는 ‘정체’ 단계가 많아지며, 평균 속도가 15km/h 이하가 된다.  통학을 하는 학생으로서 개인적으로도 교통 체증을 많이 느낀다. 비교적 교통체증이 덜한 지하철도 출/퇴근 시간이나 점심시간에는 지하철 안이 사람으로 가득 차게 되는 경우가 많으며, 그로 인해 열차가 지연된 적도 많았다. 하지만 고작 30분 차이로 도로가 한산해 지는 경우도 많으며, 우회해서 가는 길은 계속 원활할 때도 있다. 이런 상황은 서울의 교통 체증이 워낙 가변적이다 보니 요일이나 시간에 따라 교통 정체의 정도가 많이 차이가 나 예측 불가능하며 운에 따른 거라고 생각하기도 한다.  하지만 분명, 교통 체증이 심해질 때의 패턴이 있을 거라고 생각을 했고, 그 패턴을 파악한다면 막히는 길을 우회해서 가거나 아니면 조금 일찍, 또는 조금 늦게 출발해 교통 체증을 조금이라도 피할 수 있다는 생각을 했다. 그래서 그 패턴을 파악하고자 개인적인 이동 경로를 기록하고 쉽게 찾아볼 수 있는 기능을 만들고자 한다. 추가적으로 네비게이션과 다르게 골목길 같이 걸어다니는 길도 따로 기록할 수 있게 하여 차별점이 있다고 생각한다. |

|  |
| --- |
| **5. 본론 (1장 이내)**    그림 2 프로그램 개요도  그림 2 프로그램 실행 화면(예상)  프로그램에서 지도를 다루기 위해선 외부 라이브러리가 필요하다. python에서는 folium 라이브러리가 그 역할을 도와준다. 또, 시간을 다루기 위해 time 라이브러리를 사용하며 마지막으로 db와 연결을 도와줄 라이브러리로 pycopg2를 이용한다. DB는 MYSQL을 이용해 구현할 예정이다.  python을 이용해 전체적인 프로그램의 틀과 로직을 구성할 예정이다. 기록을 선택할 시 프로그램은 다음과 같은 흐름으로 전개된다.  이후 측정한 데이터를 DB로 보내는 과정 사이 의미 있는 데이터만 골라내기 위한 로직도 추가한다. 예를 들어, 같은 자리에 10분이상 머물러 있거나, 같은 자리를 계속해서 도는 행동 등, 교통 체증으로 생각되지 않는 움직임의 경우 이동 소요 시간에 포함되지 않도록 사전 처리를 한다. 각 작업의 우선순위의 경우 먼저 folium을 이용한 지도 관련 부분을 최우선으로 한다. 이 작업이 완료 되어야 이후 데이터의 후처리나 DB에 삽입하는 등 프로그램이 진행되기 때문이다. 이후 time라이브러리를 이용해 데이터에 의미를 불어 넣는 작업을 거친 후, 최종적으로 DB를 설계하고 구성한 후 최종적으로 데이터를 삽입한다. github를 이용해 협업을 하는 만큼 신속하고 효율적으로 서로의 피드백을 할 수 있는 환경을 만든다. 필요에 따라 각 작업에 추가적인 인력을 투입해 유동적으로 개발할 수 있도록 할 생각이다. |

|  |
| --- |
| **6. 결론**  교통 체증의 근본적인 원인을 해결하진 못해도 개개인이 느끼는 교통 정체를 완화할 수 있는 한 가지의 방안이다. 자신이 즐겨 가는 길을 기록해두고 하나의 목적지에 한 가지의 길이 아닌 유동적으로 다른 길들을 택하면서 이동 소요 시간을 최소화하는 기능을 만들 예정이다.  향후 folium의 사용법을 숙지하고 MySQL을 이용해 DB를 설계하는 등 추가적인 공부가 필요한 부분에서 노력할 것이다. |

**7. 출처**

[1] Rob Story, “Folium”, <https://python-visualization.github.io/folium/latest/>

[2] 서울교통 빅데이터 플랫폼, https://t-data.seoul.go.kr/