알고리즘 기반 최적 경로 알고리즘 프로토타입 개발

최준원

프로젝트 개요

본 프로토타입은 가야하는 모든 정거장의 위도와 경도를 제공 받고 각 정거장의 대기인원을 대략적으로 알 수 있을 때 각 정거장의 **최단 경로 제시** 및 **최소 버스 이용량**을 제공한다.

1. 시스템 아키텍처 개요

본 시스템은 스프링 부트(Spring Boot)를 기반으로 한 **REST API 서버**와, 결과를 시각화하는 **동적 웹 페이지**로 구성각 기능은 **역할과 책임에 따라 여러 서비스 클래스로 분리**

- **주요 기술 스택:**
    - **백엔드:** Java, Spring Boot
    - **최적화 엔진:** Google OR-Tools
    - **외부 API:** Kakao Mobility API (길찾기)
    - **프론트엔드:** HTML, JavaScript, Kakao Maps API

핵심 처리 흐름:

1. 정류장 데이터 로딩 (StopDataService)
2. 수요가 많은 정류장을 가상 정류장으로 분할 (StopDataService)
3. 실시간 이동 시간 행렬 생성 (API 호출 및 캐싱) (KakaoApiService)
4. 최적 경로 및 필요 버스 대수 계산 (RouteOptimizationService with OR-Tools)
5. 계산 결과를 DTO로 가공 및 도착 시간 역산 (SolutionFormatterService)
6. 최종 결과를 JSON으로 API 응답 (RouteController)
7. 웹 브라우저에서 데이터 수신 및 동적 지도 시각화 (index.html)

2.프로토 타입 시연

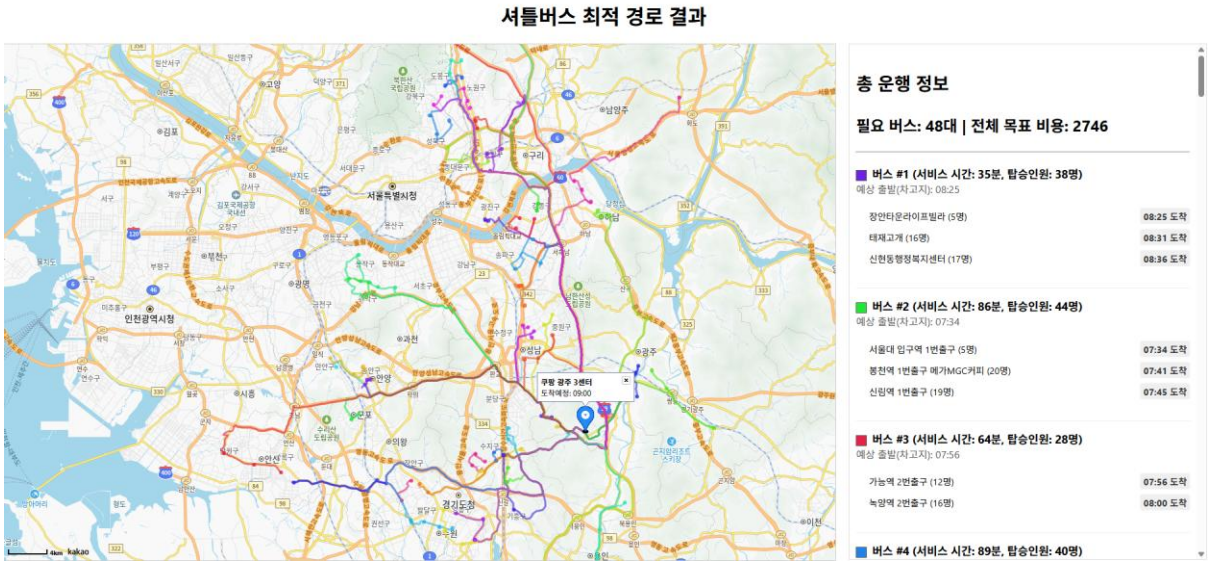
변수:

- 버스의 수용 인원: 45명
- 버스 정거장 사이의 권장 이동시간: 20분
- 정거장 대기인원: Random(20)+5
- 첫 경유지부터 도착지까지의 제한 시간: 100분
- 경유지 리스트(120개):예시

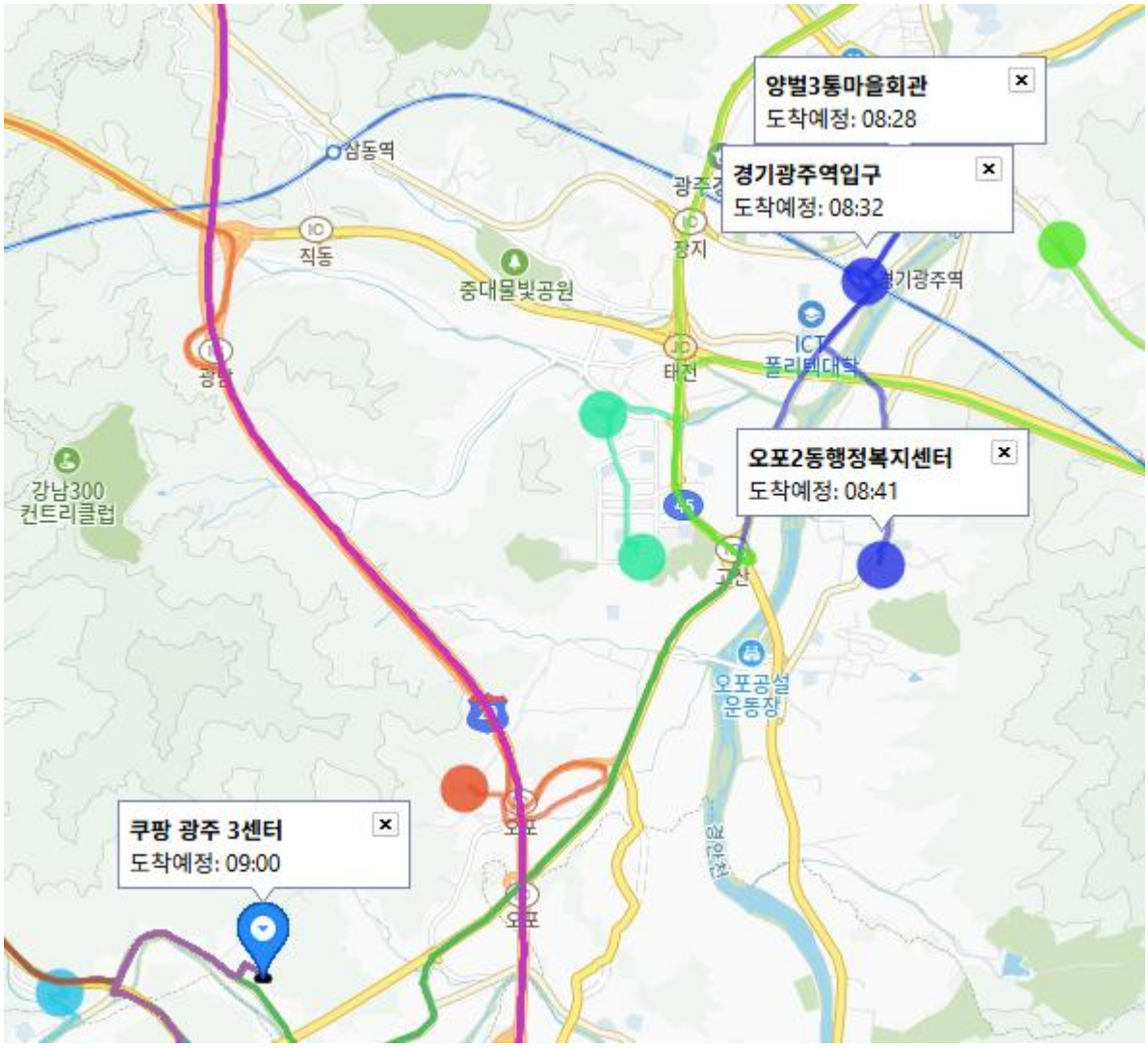
장소	위도	경도
오리역 2번 출구	37.34000853	127.1094074
미금역 3번 출구	37.34984768	127.1095119

시연 결과

전체 사진



세부 사진 1

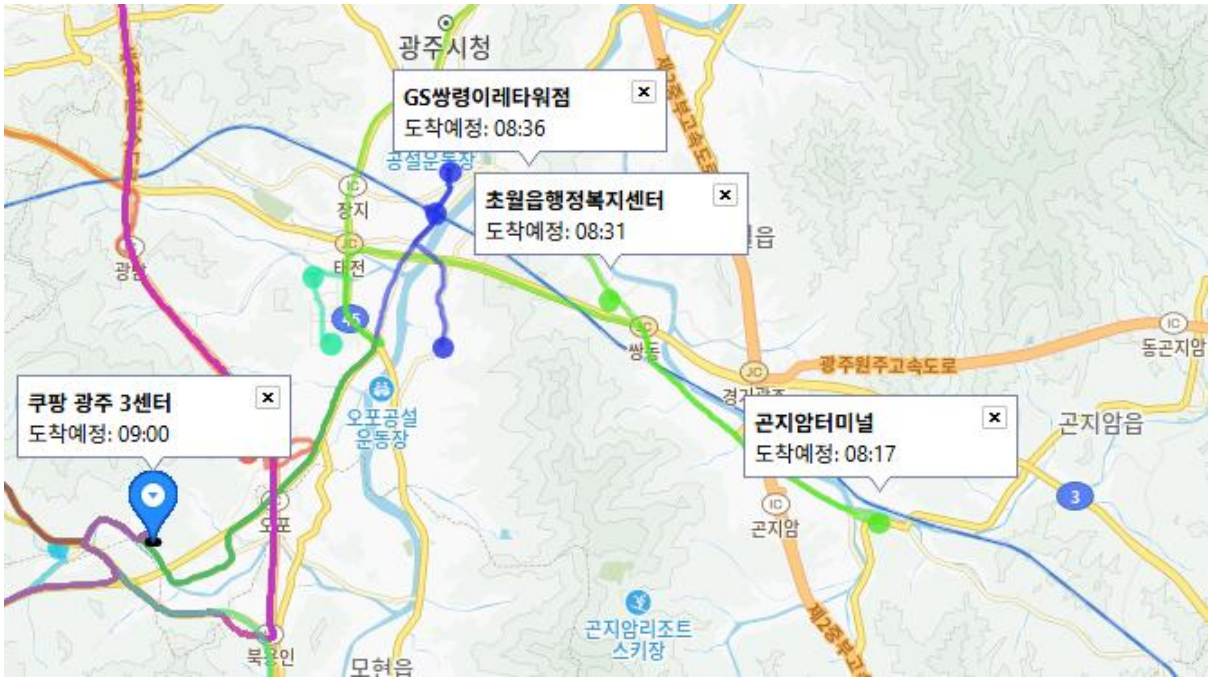


버스 #30 (서비스 시간: 32분, 탑승인원: 41명)

예상 출발(차고지): 08:28

양벌3통마을회관 (20명)	**08:28 도착**
경기광주역입구 (6명)	**08:32 도착**
오포2동행정복지센터 (15명)	**08:41 도착**

세부사진 2

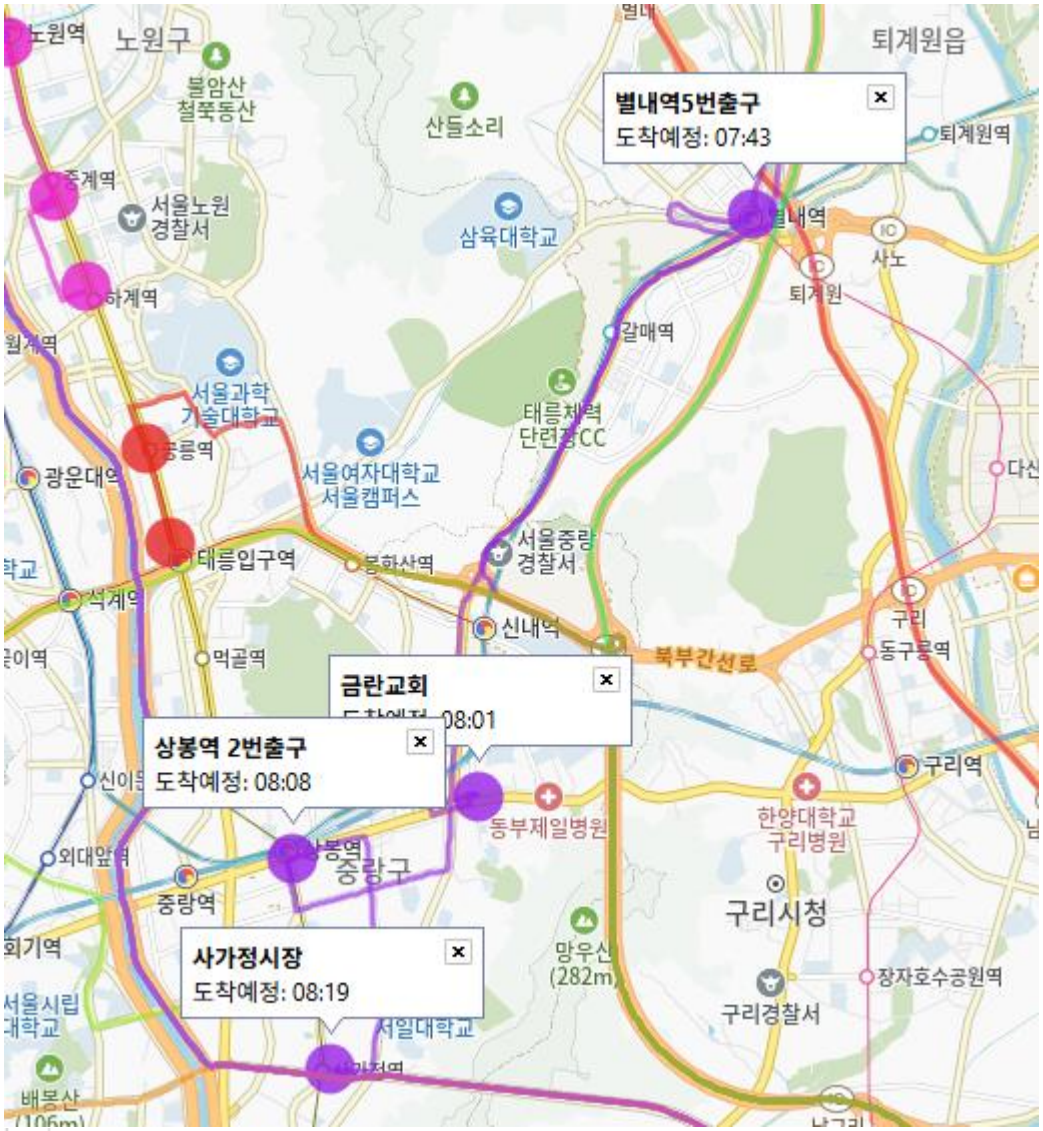


버스 #10 (서비스 시간: 43분, 탑승인원: 44명)

예상 출발(차고지): 08:17

곤지암터미널 (11명)	**08:17 도착**
초월읍행정복지센터 (14명)	**08:31 도착**
GS쌍령이레타워점 (19명)	**08:36 도착**

세부사진 3



버스 #14 (서비스 시간: 77분, 탑승인원: 41명)

예상 출발(차고지): 07:43

별내역5번출구 (5명)	**07:43 도착**
금란교회 (22명)	**08:01 도착**
상봉역 2번출구 (5명)	**08:08 도착**
사가정시장 (9명)	**08:19 도착**

테스트 설명

경유지를 200M 원으로 표시하여 가시성을 높여 테스트

오전 9시를 도착기점으로 설정하여 카카오API를 통해서 각 정류장의 시간 요소를 역산하여 예상 도착시간을 도입

경유지 사이의 시간은 20분으로 제한

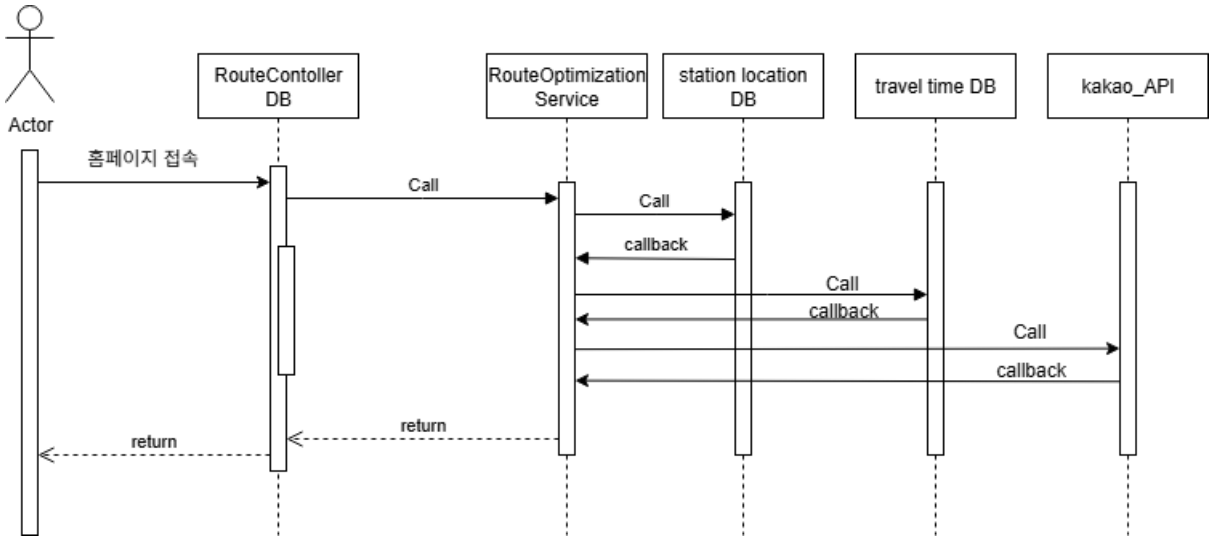
서비스 시간(첫 경유지부터 도착시간까지)는 100분으로 제한

버스 용량 45명

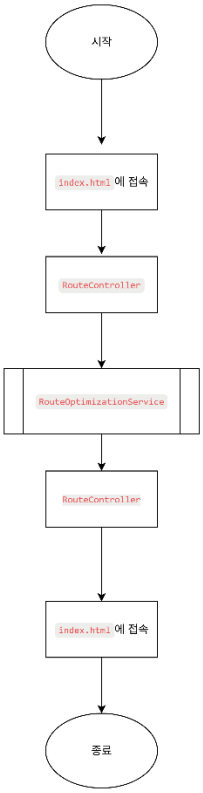
각 경유지 대기인원 Random(21)+5

3. 세부 알고리즘 설명

프로토 타입 시퀀스 다이어그램



프로토 타입 알고리즘 순서도


시작
index.html 에 접속
RouteController
RouteOptimizationService
RouteController
index.html 에 접속
종료


RouteOptimizationService
탑승 가능 인원수가
버스 탑승 인원수 보다 많습니까?
yes
No
정거장을 가상 정거장으로 분할
모든 정거장에 이동시간이 지정 돼 있습니까?
no
yes
KakaoApiService
RouteOptimizationService
SolutionFormatterService
종료

StopDataService 기술 명세

1. 목적 및 역할

StopDataService는 시스템의 데이터 준비 계층으로서, 외부 파일(location.csv)로부터 원본 정류장 데이터를 로드하고, 이를 OR-Tools 최적화 엔진이 사용할 수 있는 형식으로 가공하는 모든 책임을 가진다.

본 서비스의 핵심 역할은 물리적 정류장(PhysicalStop) 목록을 생성하고, 각 정류장의 수요(demand)와 버스의 최대 수용량(vehicleCapacity)을 비교하여 가상 정류장(VirtualStop) 목록으로 변환하는 것이다.

2. 알고리즘 처리 순서

RouteOptimizationService가 getVirtualStops(vehicleCapacity) 메소드를 호출하면, 아래의 순차적인 알고리즘에 따라 작업을 수행한다**.**

단계 1: 원본 데이터 로딩 (Loading Physical Stops)

1. 가장 먼저, 내부 헬퍼 메소드인 createPhysicalStopsFromCsv()를 호출하여 원본 정류장 데이터의 로딩을 시작한다.
2. 파일 접근: ClassPathResource를 사용하여 src/main/resources/ 경로에 있는 location.csv 파일을 스트림(Stream) 형태로 불러온다.
3. 데이터 파싱 (Parsing): BufferedReader를 이용해 파일을 한 줄씩 읽어 들인다.
    - 각 라인은 쉼표(,)를 구분자로 하여 분리(split)한다.
    - 분리된 데이터 배열에서 정류장 이름, 위도, 경도를 추출한다.
    - ID는 ST_1, ST_2와 같이 동적으로 생성하며, 수요(demand)는 지정된 범위 내에서 랜덤 값을 할당한다.
    - "검색실패" 문자열이 포함되거나, 숫자 변환이 불가능한 라인은 오류 로그를 출력하고 건너뛰어 처리의 안정성을 보장한다.
4. 객체 생성: 파싱된 데이터를 사용하여 PhysicalStop 객체를 생성하고, 이들을 List<PhysicalStop>에 순차적으로 추가한다. (이때, 리스트의 첫 번째 요소로는 항상 고정된 차고지(Depot) 정보가 먼저 추가된다.)
5. 최종적으로, 완성된 PhysicalStop 목록을 반환한다.

단계 2: 가상 정류장 변환 (Transforming to Virtual Stops)

1. getVirtualStops 메소드는 1단계에서 생성된 physicalStops 목록과, 파라미터로 받은 vehicleCapacity(버스 정원)를 사용하여 가상 정류장 생성을 시작한다.
2. 수요 분할 로직 (Demand Splitting): physicalStops 목록을 for문으로 순회하며 각 정류장을 검사한다.
    - 수요 ≤ 정원: 만약 정류장의 총수요(demand)가 버스 정원(vehicleCapacity)보다 작거나 같으면, 해당 PhysicalStop의 정보를 그대로 사용하여 VirtualStop 객체 한 개를 생성하고 리스트에 추가한다.
    - 수요 > 정원: 만약 수요가 정원을 초과하면, while 반복문을 실행하여 수요가 0이 될 때까지 다음 로직을 반복한다.
        - 현재 남은 수요(remainingDemand)와 버스 정원 중 더 작은 값을 currentDemand로 결정한다.
        - 이 currentDemand를 수요로 가지는 새로운 VirtualStop 객체를 생성한다. 이때, 이름 뒤에 -1, -2와 같은 접미사를 붙여 구분한다.
        - 남은 수요에서 방금 처리한 수요를 빼고, 모든 수요가 처리될 때까지 반복한다.

단계 3: 최종 목록 반환

모든 PhysicalStop에 대한 변환 작업이 완료되면, 최종적으로 완성된 List<VirtualStop>을 RouteOptimizationService에 반환하며 모든 작업을 종료한다.

KakaoApiService 기술 명세

1. 목적 및 역할

KakaoApiService는 시스템의 외부 통신 계층으로서, 카카오모빌리티 API와의 모든 상호작용을 전담한다. 본 서비스의 핵심 역할은 VirtualStop 객체 목록을 입력받아, 각 정류장 쌍(pair) 간의 실제 이동 시간으로 구성된 2차원 시간 행렬(Time Matrix)을 생성하여 반환하는 것이다.

2. 주요 메소드 및 알고리즘 처리 순서

2.1. createTimeMatrixFromApi(List<VirtualStop> stops)

- VirtualStop 목록을 받아, 캐싱 전략을 사용하여 2차원 시간 행렬을 생성하는 메인 메소드이다.
- **알고리즘 순서:**
    1. 캐시 로딩: loadDurationsFromCache를 호출하여 cache/timeMatrix_name_cache.json 파일에서 기존 캐시를 로드한다.
    2. 경로 쌍 생성 및 캐시 조회: 입력받은 VirtualStop 목록을 기준으로 모든 경로 쌍에 대해, 사전순으로 정렬된 이름으로 정규화된 키(정류장A_정류장B)를 생성하고, 해당 키가 캐시에 존재하는지 확인한다.
    3. 조건부 API 호출:
        - Cache Hit (캐시에 키가 존재): API를 호출하지 않고 다음 경로 쌍으로 넘어간다.
        - Cache Miss (캐시에 키가 없음): getDurationInMinutes를 통해 실제 카카오 길찾기 API를 호출하고, 결과를 메모리의 캐시 Map에 추가한다.
    4. 캐시 파일 저장: API 호출로 인해 캐시가 업데이트된 경우, saveDurationsToCache를 호출하여 전체 캐시 Map을 JSON 파일에 덮어쓰기하여 저장한다.
    5. 최종 행렬 생성 및 반환: buildMatrixFromCache를 호출하여, 완성된 캐시 Map을 기준으로 최종 long[][] 타입의 2차원 시간 행렬을 구성하여 반환한다.

2.2. getDetailedPath(VirtualStop origin, VirtualStop destination)

- 두 지점 간의 실제 도로를 따라가는 상세 경로 좌표 목록을 조회하여 반환한다. 이 데이터는 프론트엔드의 지도 시각화에 사용된다.
- 알고리즘 순서:
    1. HTTP 요청: RestTemplate을 사용하여 https://apis-navi.kakaomobility.com/v1/directions 엔드포인트로 GET 요청을 전송한다. 요청 URL에는 출발지와 도착지의 위도/경도 좌표가 포함된다.
    2. 응답 처리: 응답받은 JSON 데이터에서 routes[0].sections[].roads[].vertexes 경로에 있는 좌표 목록을 파싱한다.
    3. 데이터 가공: vertexes 배열은 [경도, 위도, 경도, 위도...] 순서의 1차원 배열이므로, 이를 (위도, 경도) 쌍을 가지는 LatLngDto 객체의 리스트로 변환한다.
    4. 결과 반환: 가공된 상세 경로 좌표 리스트(List<LatLngDto>)를 반환한다. API 호출 실패 시에는 빈 리스트를 반환한다.

2.3. getTimeToDepot(List<PhysicalStop> allStops)

- RouteOptimizationService의 사전 분리 로직을 지원하기 위해, 모든 물리적 정류장에서 차고지까지의 이동 시간을 미리 계산하여 Map 형태로 반환한다.
- 알고리즘 순서:
    1. 입력받은 PhysicalStop 목록에서 차고지(depot) 정보를 추출한다.
    2. 목록의 나머지 모든 정류장에 대해 for문을 순회하며, getDurationInMinutes 메소드를 호출하여 해당 정류장과 차고지 사이의 이동 시간을 계산한다.
    3. 계산된 결과를 Map<String, Long> (key: 정류장 ID, value: 이동 시간)에 저장하여 반환한다.

RouteOptimizationService.java

1. **목적 및 역할:**

   Google OR-Tools 엔진을 직접 실행하는 가장 핵심적인 역할을 수행

2. **알고리즘 처리 순서**

   - **데이터 준비:** StopDataService를 호출하여 원본 PhysicalStop 목록을 로드합니다.
   - **최적화 실행:**
     1. runOptimization은 대상 정류장들만으로 구성된 부분 시간 행렬(subTimeMatrix)을 생성하고, Google OR-Tools에 모든 제약조건(비용, 서비스 시간, 용량)을 설정한 뒤, solveWithParameters()를 실행하여 최적의 해답(Assignment 객체)을 찾습니다.
   - **결과 병합 및 반환:**
     1. 최적화가 완료된 Assignment 객체와, 처음에 미리 분리해 둔 '단독 운행 그룹' 목록을 모두 SolutionFormatterService에 전달하여, 최종 결과물(RouteSolutionDto)로 가공해달라고 요청하며 모든 프로세스를 마무리

SolutionFormatterService.java

1. **목적 및 역할:**

   OR-Tools가 계산한 복잡하고 추상적인 결과(Assignment 객체)와 사전 처리된 데이터를 받아, 최종적으로 프론트엔드가 사용하기 좋은 **RouteSolutionDto 형태로 가공하고 포장**하는 책임을 가집니다.

2. **알고리즘 처리 순서 (formatFinalSolution 메소드):**

   - **최적화 결과 포장:** formatOptimizedRoutes를 호출하여, OR-Tools가 계산한 '서비스 가능 그룹'의 경로를 DTO 리스트로 변환합니다.
     1. 이 과정에서 각 버스의 **총 운행 시간**을 먼저 계산하고, 이를 바탕으로 "오전 9시 최종 도착"을 가정하여 **필요 출발 시각**과 **경유지별 예상 도착 시각**을 역산합니다.
   - **최종 병합 및 정리:**
     1. 전체 운행 버스 대수를 기준으로, generateDistinctColors를 호출하여 **다양한 경로 색상**을 동적으로 생성합니다.
     2. 최종적으로 합쳐진 경로 목록에 **버스 ID를 1번부터 순서대로 다시 부여**하고, 생성된 색상을 할당하여 완벽한 일관성을 보장합니다.
     3. 이 모든 정보를 RouteSolutionDto에 담아 반환합니다.

사용한 라이브러리

분류	라이브러리 / 프레임워크
백엔드 프레임워크	**Spring Boot**
최적화 엔진	**Google OR-Tools**
외부 API (서버)	**Kakao Mobility API**
외부 API (클라이언트)	**Kakao Maps JavaScript API**
테스트 프레임워크	**JUnit 5**
JSON 처리	**Jackson**

개선하면 좋을 사항

1. **프론트에서 입력을 하며 그 입력에대해 최적의 경로를 다시 뽑을 수 있도록 하는 기능 제공**

현재는 프론트에서 입력을 줄 수 없지만 추후에 추가 가능하다

2. **추천 경로를 3가지 정도 뽑을 수 있도록 구성**

최대 경로 시간을 기준으로 버스를 최소로 줄이는 방안, 버스를 좀 늘리고 경로 이동시간을 늘리는 방안등을 고려하여 3가지정도 제시 가능할 수 있음

3. **데이터베이스 적용**

현재는 테스트를 위해서 csv파일을 사용하지만 이는 DB로 적용 가능하다

4. **경로 완벽 안내**

현재 카카오 api에서는 네비게이션 기능을 제공하지 않지만 추후 연구를 통해 가능할 수도 있을거라고 생각함

5. **api호출 시간문제**

   api 호출 시간은 굉장히 오래걸림(정류장100개당 10000번의 호출이 필요)따라서 캐시 기능을 이용하여 이를 최소화 시켰으나 여전히 오래걸림

6. **계산 불가 가능성**

사용자가 원하는 조건이 많으면 많을수록 연산량은 늘어나게 되고 이에 따라 최종 결과가 가능성 없는 조건일 수도 있음 하지만 이를 미연에 방지하기 힘듬

7. **사용자가 원하는 경로 이동시간보다 정거장이 먼 경우에 처리 알고리즘의 부재**

현재 사용자가 원하는 경로 이동시간보다 정거장이 먼 경우에는 알고리즘 연산이 되지 않음을 확인함 따라서 입력에 제한을 주거나 다른 입력을 주도록 유도하도록 해야함

8. **테스트의 부재**

테스트는 매우 중요하지만 가질 수 있는 데이터셋이 매우 부족하여 테스트하기 힘듬

9. **각 정류장의 일반적인 대기시간 추가**

어쩔 수 없이 각 정류장의 대기시간은 불가피하다. 따라서 해당 대기시간에 대한 알고리즘이 추가적으로 필요하다.

부록(테스트 데이터)

장소	위도	경도
오리역 2번 출구	37.34000853	127.1094074
미금역 3번 출구	37.34984768	127.1095119
대원사거리	37.43335276	127.1599747
AK프라자	37.26555243	127.00069
임광아파트	37.25733946	127.0385703
장안타운라이프빌라	37.37353201	127.143292
태재고개	37.36053762	127.1475206
신현동행정복지센터	37.35965437	127.1588428
오포롯데캐슬	37.3507999	127.1641763
오포베르빌아파트	37.34588516	127.1776431
신명아파트	37.64662749	127.3038265
수지동부아파트101동	37.32784455	127.0937355
수지이마트	37.31981961	127.0833553
LG빌리지5차	37.32656626	127.0728651
성복동행정복지센터	37.3164878	127.0690422
용인시평생학습관	37.32036001	127.0955641
용인포은아트홀	37.32560522	127.1047978
내대지 우미이노스빌 6차	37.33116144	127.124756
신림역 1번출구	37.48414757	126.9302045
봉천역 1번출구 메가MGC커피	37.48201014	126.9432087

서울대 입구역 1번출구	37.48089725	126.9531476
낙성대역 3번출구(국민은행)	37.47681832	126.9631453
사당역 1번출구	37.47625601	126.9823749
사가정시장	37.58114332	127.0894014
면목역 1번출구	37.5895619	127.0873984
상봉역 2번출구	37.59632945	127.0859003
금란교회	37.60071102	127.1027421
녹양역 2번출구	37.75944826	127.0424428
가능역 2번출구	37.74845569	127.0445357
의정부역 동부광장	37.73732394	127.0475626
회룡역5번출구	37.72284905	127.0472993
별내역5번출구	37.64258435	127.1276952
마들역 6번출구	37.66490268	127.057599
노원역 8번 출구	37.65514375	127.060198
중계역 4번출구	37.64400538	127.064404
하계역 6번출구	37.63744456	127.0673554
공릉역 4번출구	37.62581867	127.0727031
태릉입구역 3번출구	37.61899676	127.0750071
수유역 8번출구 마을	37.64040447	127.0258578
미아역 8번출구 마을	37.63052597	127.0218655
미아사거리역 6번출구	37.61400732	127.0298559
월곡역 5번출구	37.60188544	127.0416623

쌍문역2번출구	37.64805429	127.0345947
도봉보건소	37.65791512	127.0388445
방학역1번출구	37.66890886	127.04408
도봉역1번출구	37.68040477	127.0456172
홍파복지원	37.68048633	127.0569491
회기역 1번 출구	37.58992257	127.0576092
답십리역 5번 출구	37.56631865	127.053086
장한평역 5번 출구	37.56115897	127.0650344
군자역 5번 출구	37.55683416	127.0797774
자양사거리 스타벅스	37.53700021	127.084049
광진구의회	37.53627672	127.0877548
명일역 2번 출구	37.55172782	127.1439154
굽은다리역 1번출구	37.54551239	127.1427153
길동역 1번 출구	37.53792056	127.1397137
천호역 9번 출구	37.53825438	127.1231156
풍납현대아파트	37.52938744	127.1189038
둔촌오륜역 2번 출구	37.51910012	127.1384569
마천사거리	37.49795821	127.1465511
가락시장역 2번출구	37.49288499	127.1180883
장지역.가든파이브	37.47800469	127.1217844
동서울대학교	37.46016725	127.1287181
태평역 수정경찰서	37.44221227	127.1266804

정관장 모란역점	37.44047444	127.1322269
야탑역 3번 출구	37.41142281	127.1282732
장지역.가든파이브라이프동	37.47779072	127.1247913
남한산성입구역 3번 출구	37.45190323	127.1596819
성남우체국	37.44236439	127.149905
MD아파트	37.43696201	127.1387575
성남중원우편취급국	37.43102904	127.1362473
동원SK삼거리	37.41281051	127.1438186
GS쌍령이레타워점	37.40171753	127.2717783
보건소	37.50572903	126.9411244
공설운동장	37.40532631	126.9464811
경기광주역입구	37.39894381	127.2532827
태전동성원아파트	37.38908636	127.228664
힐스테이트태전2차에듀포레	37.37842394	127.2322232
오포초등학교	37.36112169	127.2154985
곤지암터미널	37.34956506	127.3417685
초월읍행정복지센터	37.38524824	127.2879607
양벌3통마을회관	37.40559382	127.2563998
오포2동행정복지센터	37.37773626	127.2546475
하남검단산역.창우초교	37.54066347	127.2238261
하남시청	37.53927457	127.2148606
호반써밋에듀파크	37.5452254	127.2110054

하남풍산역.풍산아이파크5단지	37.55311247	127.2034521
미사역5번출구	37.56337935	127.1932413
미사강변스타힐스	37.5714838	127.1847996
이차돌 안양석수점	37.41261113	126.9095974
안양 미문교회	37.40478954	126.9159227
비산사거리.이마트	37.39837207	126.9353324
KT 안양지사	37.39341442	126.9541326
범계사거리	37.38922765	126.9482632
평촌 홈플러스	37.39331561	126.9504151
신환사거리 한국타이어 군포점	37.36258638	126.9410615
산본역 4번출구	37.35769149	126.933313
안산역 1번 출구 파리바게뜨	37.32739439	126.7894981
고잔역 1번출구	37.31687892	126.8233887
한대앞역 1번출구	37.30995547	126.8538507
상록수체육관	37.30096602	126.8686031
두산기술원	37.31116479	127.0800066
병점역사거리	37.20790912	127.0350349
오산칠갑산묵은지삼겹살	37.15779437	127.0577975
NH농협은행 오산시지부	37.14877954	127.0690763
용인터미널	37.23287601	127.2097025
럭키마트 분식코너	37.25844096	127.2129331
신원아파트	37.62582776	127.0431869

대웅경영개발원	37.2825627	127.2406709
만석공원테니스장	37.30249143	127.000758
수원KT위즈파크	37.29973025	127.009772
교육청사거리	37.29235863	127.0191091
인계선경아파트	37.27620431	127.0365233
kt플라자 동수원점	37.27311738	127.0510619
기흥역1번출구	37.27660528	127.1161597
구성역.연원마을LG	37.30158813	127.1078245
보정고등학교	37.31390181	127.109647
성균관대역 1번출구	37.30032032	126.972389
화서역 4번출구	37.28484324	126.9894563
수원역 4번출구	37.26753472	126.9998647
매교역4번출구	37.26556949	127.0150956
수원시청역 1번출구	37.26128609	127.0323315
영통구청	37.25960358	127.0466248
망포역 1번출구	37.24565194	127.0585295
살구골현대아파트	37.24971924	127.0716684
영통입구	37.26820973	127.081145