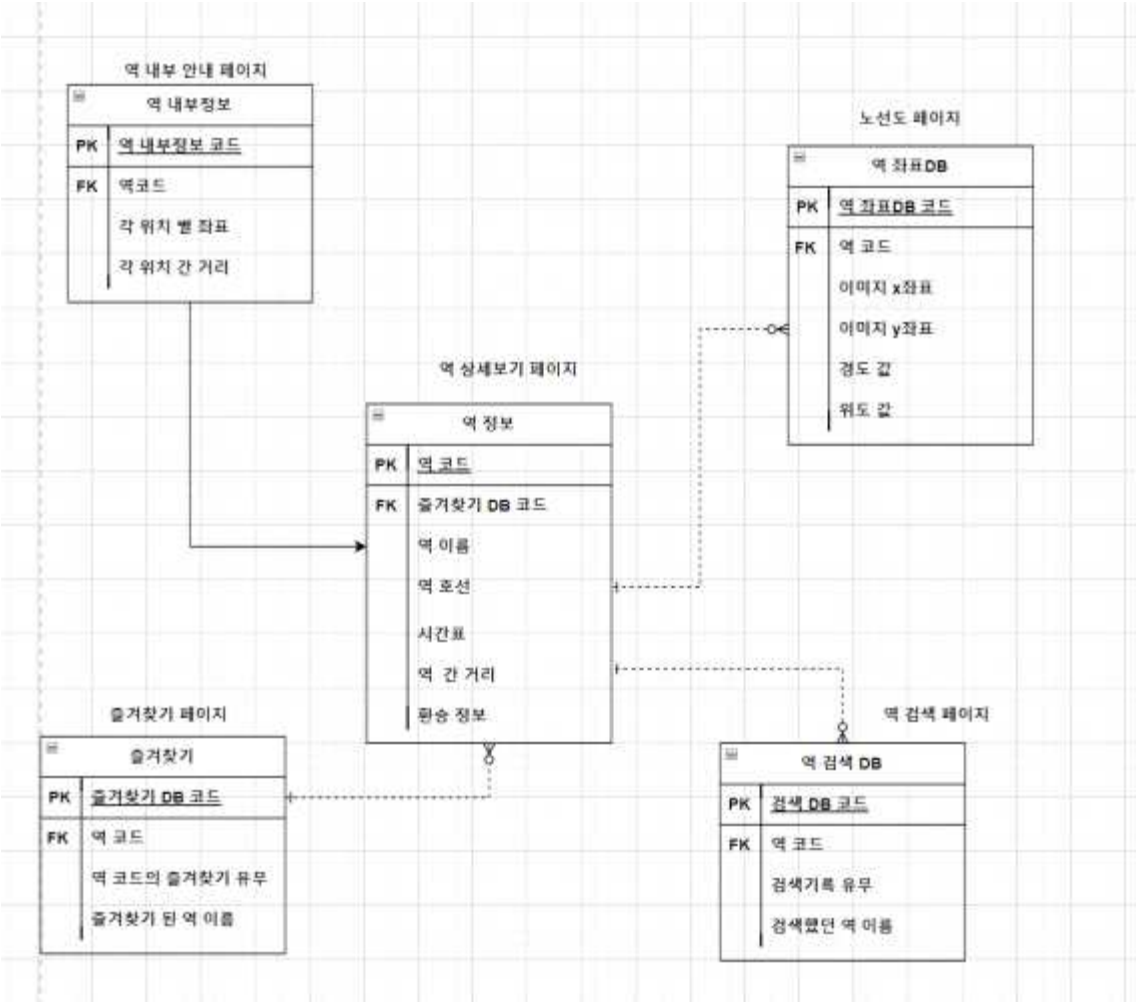
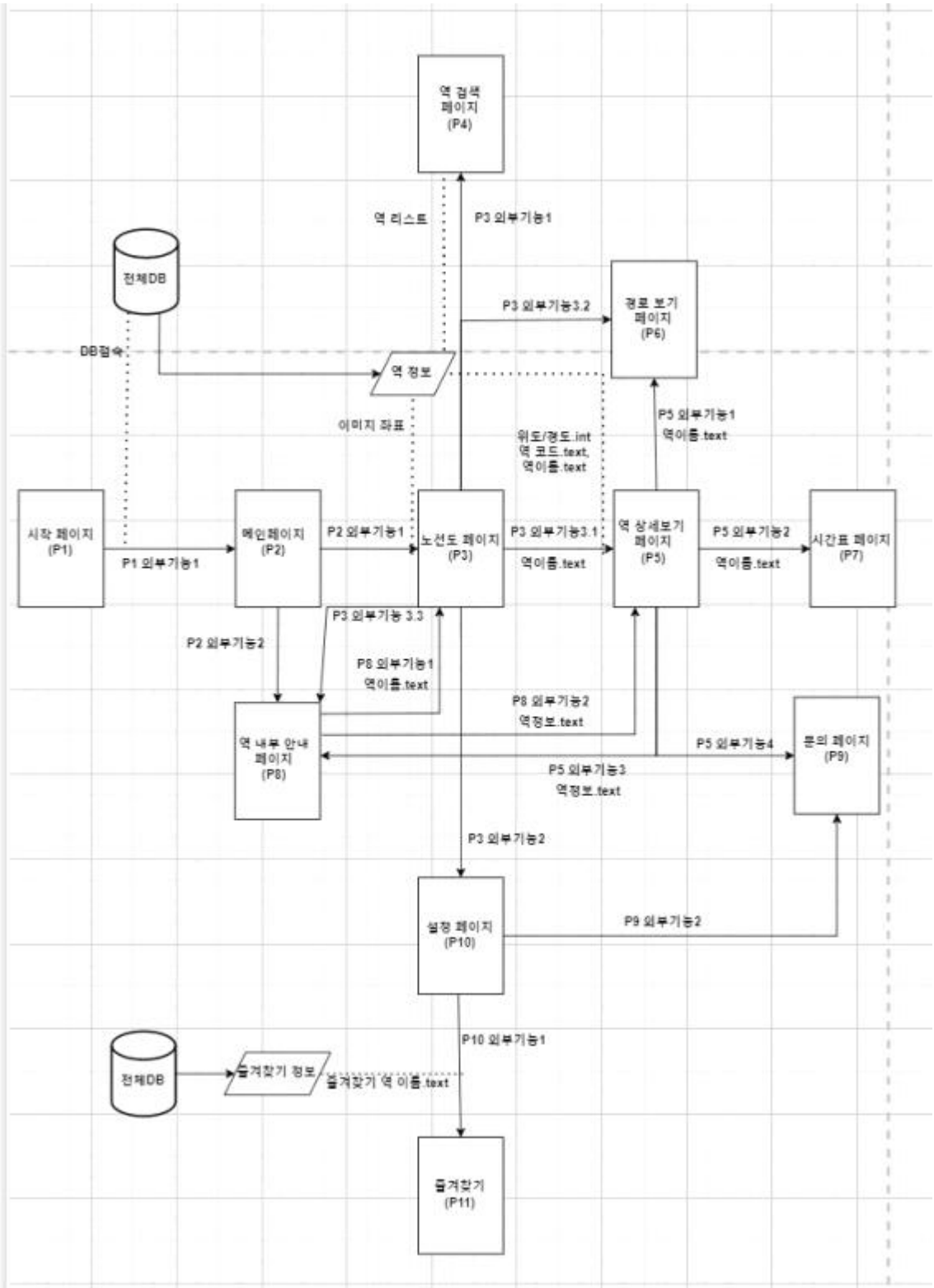
캡스톤 5조 보고서

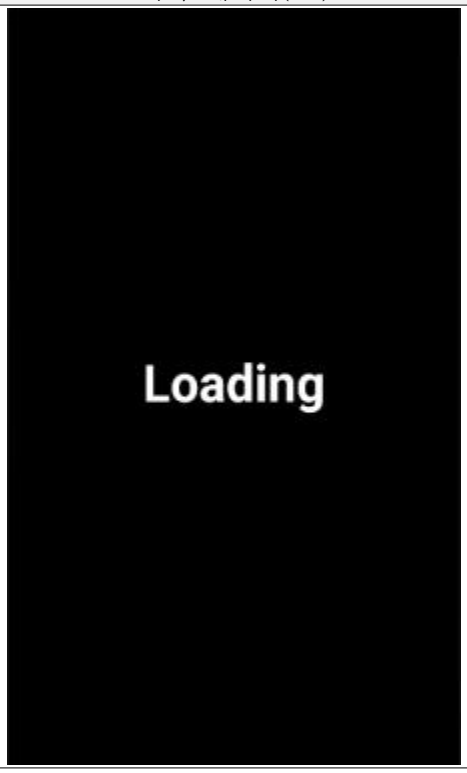
<table>
<tr><td>프로젝트명</td><td colspan="6">지하철 역 내부 네비게이션 (station inner navigation)</td></tr>
<tr><td>프로젝트 설명</td><td colspan="6">다익스트라(Dijkstra) 알고리즘을 사용하여
역과 역 경로 및 역 내부 경로를 안내해주는 지하철 어플 개발</td></tr>
<tr><td>프로젝트 기간</td><td colspan="6">2022.3.8. ~ 2022.6.7 / 2022.9.~</td></tr>
<tr><td rowspan="10">조원 및 담당업무</td><td>이름</td><td>강홍준</td><td>학과</td><td>소프트웨어학과</td><td>학번</td><td>20161267</td></tr>
<tr><td colspan="6">기획, 데이터 수집, 레이아웃/내부기능 개발</td></tr>
<tr><td>이름</td><td>김경한</td><td>학과</td><td>소프트웨어학과</td><td>학번</td><td>20161224</td></tr>
<tr><td colspan="6">기획, 데이터 수집, UI/UX 디자인</td></tr>
<tr><td>이름</td><td>김도영</td><td>학과</td><td>소프트웨어학과</td><td>학번</td><td>20161225</td></tr>
<tr><td colspan="6">기획, 데이터 수집, UI/UX 디자인</td></tr>
<tr><td>이름</td><td>박준호</td><td>학과</td><td>소프트웨어학과</td><td>학번</td><td>20151643</td></tr>
<tr><td colspan="6">기획, 데이터 수집, UI/UX 디자인</td></tr>
<tr><td>이름</td><td>송택관</td><td>학과</td><td>소프트웨어학과</td><td>학번</td><td>20161288</td></tr>
<tr><td colspan="6">기획, 레이아웃/내부기능 개발</td></tr>
<tr><td>개발 목적 및 동기</td><td colspan="6">종종 낯선 지하철 역을 방문하게 될 때 목적지에 잘 도달하는 사람도 있지만, 그렇지 않은 사람도 있기 때문에 그것을 해소시키기 위해 개발</td></tr>
<tr><td>기대효과</td><td colspan="6">길을 헤매다가 시간에 늦는 경우를 방지할 수 있고,
역 내부 경로의 거리 및 시간을 앎으로써 좀 더 시간 관리를 철저히 할 수 있다.</td></tr>
<tr><td>개발 목표 및
개선 부분</td><td colspan="6">1~9호선 데이터로 기본 지하철 어플의 기능을 넣고 3가지 역에 대해 내부 안내를 할 수 있도록 하는 것이 목표였으나
오류처리/예외처리가 되지 않았고, 역 내부 안내도 만족스럽지 않은 방식으로 실행되기 때문에 앞으로 오류처리, 데이터 추가, 역 내부 안내 개선 예정</td></tr>
</table>

DB설계


역 내부 안내 페이지
역 내부정보
PK
역 내부정보 코드
FK
역코드
각 위치 별 좌표
각 위치 간 거리
노선도 페이지
역 좌표DB
PK
역 좌표DB 코드
FK
역 코드
이미지 x좌표
이미지 y좌표
경도 값
위도 값
역 상세보기 페이지
역 정보
PK
역 코드
FK
즐겨찾기 DB 코드
역 이름
역 호선
시간표
역 간 거리
환승 정보
즐겨찾기 페이지
즐겨찾기
PK
즐겨찾기 DB 코드
FK
역 코드
역 코드의 즐겨찾기 유무
즐겨찾기 된 역 이름
역 검색 페이지
역 검색 DB
PK
검색 DB 코드
FK
역 코드
검색기록 유무
검색했던 역 이름

기능도


역 검색 페이지 (P4)
역 리스트
P3 외부기능1
전체DB
P3 외부기능3.2
경로 보기 페이지 (P6)
DB접속
역 정보
이미지 좌표
위도/경도.int
역 코드.text,
역이름.text
P5 외부기능1
역이름.text
시작 페이지 (P1)
P1 외부기능1
메인페이지 (P2)
P2 외부기능1
노선도 페이지 (P3)
P3 외부기능3.1
역이름.text
역 상세보기 페이지 (P5)
P5 외부기능2
역이름.text
시간표 페이지 (P7)
P3 외부기능 3.3
P2 외부기능2
P8 외부기능1
역이름.text
P8 외부기능2
역정보.text
역 내부 안내 페이지 (P8)
P5 외부기능4
문의 페이지 (P9)
P5 외부기능3
역정보.text
P3 외부기능2
설정 페이지 (P10)
P9 외부기능2
P10 외부기능1
전체DB
즐겨찾기 정보
즐겨찾기 역 이름.text
즐겨찾기 (P11)

시작 페이지(P1)

내부기능 1. 3초간 로딩화면 보여준 후 이동 - postDelayed를 이용하여 3000mills 딜레이 하여 로딩화면 보여주기 **외부기능** 1. 메인페이지(P2)로 이동 - postDelayed 함수 안에 intent를 집어넣어 3초 후 메인페이지(P2)로 넘어가게 설정

메인 페이지(P2)

내부기능

1. 경로 찾기를 위한 그래프 생성
 - 그래프 초기화
 1.1. 그래프에 노드를 입력하기 위해 DB접속
  - DB subway_info의 subway_time 테이블을 참조하는 cursor_time 생성
 1.2. 그래프에 값 삽입
  - while(cursor_time.moveToNext)를 사용하여 커서의 모든 값을 삽입한다.
  (start, end, time이며 역과 역을 이어주고 역간 소요시간이 입력됨)

외부기능

1. 경로 찾기 버튼
 - 클릭 시 intent로 노선도 페이지로 이동(P3)

2. 역 내부 안내 버튼
 - 클릭 시 intent로 노선도 페이지로 이동(P3)

노선도 페이지(P3)

<클릭 전>

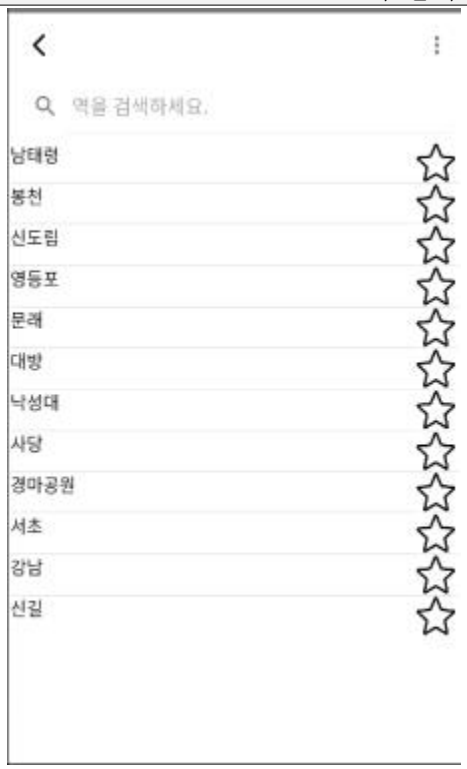
<클릭 후>

내부기능

1. 내부 버퍼에 있는 지하철 노선도 이미지 불러오기
 - 오픈소스 SubsamplingScaleImageView 사용하여 이미지 배치
 (SubsamplingScaleImageView사용으로 줌 인/아웃 및 이미지 이동 가능)
2. 특정 역 클릭시 팝업 활성화
 2.1. 터치 이벤트를 처리하기 위한 데이터 가져오기
 - 이미지의 스케일을 가져와서 TileScale에 저장한다.
 (핸드폰 기종마다 해상도가 다르기 때문에 크기에 맞춰줘야 함)
 - DB subway_info의 subway_coordinate 테이블을 참조하는 cursor_coor 생성
 2.2. 터치 이벤트 생성 - onSingleTapUp (살짝 터치)
 - 클릭된 지점의 좌표를 PointF 객체 sCoord에 저장
 - x좌표는 x_cor, y좌표는 y_cor에 저장
 - while(cursor_coor.moveToNext)를 사용하여 커서에 저장된 좌표와 비교하여
 만약 x_cor이 x1보다 크고 x2보다 작으며 y_cor이 y1보다 크고 y2보다 작다는
 조건을 만족하면 해당역에 해당하는 이름을 curStation에 저장 후 팝업을 띄운다.
 2.3. 팝업 생성
 - QuickAction 오픈소스 사용
 - ActionItem을 생성한다.(코드, 이름, 아이콘)
 - QuickAction을 생성하며 아이템을 추가한다.
3. 상단 메뉴 적용
 3.1. layout의 menu파일 적용
 - 검색 버튼과 메뉴버튼을 생성한다.

노선도 페이지(P3)
외부 기능 1. 검색 버튼 - 클릭 시 intent를 이용하여 역 검색 페이지(P4)로 이동 2. 메뉴 버튼 - 클릭 시 intent를 이용하여 설정 페이지(P10)로 이동 3. 팝업의 버튼 (경로, 상세보기, 역 내부 안내) - 클릭 시 item의 이름을 title에 저장한다. 3.1. switch(title)문 사용하여 case “상세보기”라면 intent로 curStation(현재역) 데이터를 역 상세보기 페이지(P5)로 보내고 이동한다. 3.2. case “경로”라면 intent로 curStation(현재역) 데이터를 경로 페이지(P6)로 보내고 이동한다. 3.3. case “역 내부 안내”라면 intent로 curStation(현재역) 데이터를 역 내부 안내 페이지(P8)로 보내고 이동한다. 4. 뒤로가기 버튼 - 클릭 시 finish()를 이용하여 이전 화면으로 이동

역 검색 페이지(P4)



<즐겨찾기X>

<즐겨찾기O>

내부기능

1. 즐겨찾기 된 역, 검색기록 띄워주기

1.1. 리스트 띄워주기 전 리스트 커스텀하기

- convertview를 이용하여 리스트 안에 이미지버튼(북마크 유무) 추가

1.2. DB탐색

- Search_db에서 subway_bookmark 테이블과 Search_Histroy 테이블에서 각각 즐겨찾기 된 역과 검색기록을 조회하여 리스트로 띄워준다.

2. 검색

2.1. 문자 자동완성

- searchView를 이용해 입력한 문자열에 따른 자동 완성된 리스트를 보여준다.

2.2. 입력한 문자열에 따른 차이

- 입력 문자가 0 일시 즐겨찾기와, 검색기록을 보여주고, 문자 자동완성 기능을 끈다.
- 입력 문자가 1 이상일 경우 즐겨찾기와, 검색기록을 안보여주고, 자동완성 기능을 킨다.

2.3. 검색기록 추가하기

- 자동완성 리스트에서 선택한 역을 문자열로 가져온 후 Search_Historty테이블에 추가
- 이미 Search_History 또는 subway_bookmark에 같은 문자열이 있을 경우에는 추가 하지 않음

역 검색 페이지(P4)
내부기능 2.4. 즐겨찾기 추가 - 검색기록 리스트에 빈별을 클릭 시 해당 역에 문자열을 받아와 subway_bookmark테이블에 추가 후 Search_History에서는 데이터를 삭제한다. 2.5. 즐겨찾기 해제 - 즐겨찾기 리스트에 노란 별 클릭 시 해당 역에 문자열을 받아와 subway_bookmark에서 데이터 삭제 후 Search_History에 데이터를 추가한다. 3. 전체 삭제 - 툴바에 “즐겨찾기 전체 삭제” 클릭 시 subway_bookmark와 Search_History 테이블에 데이터를 모두 삭제 **외부 기능** 1. 리스트의 역 이름 클릭 - 리스트에 아이템 getText().toString으로 값을 curStation(역 이름)에 저장하여 Intent를 이용하여 curStation에 맞는 상세보기 페이지(P5)로 이동 2. 뒤로가기 버튼 - 클릭 시 finish()를 이용하여 이전 화면으로 이동

역 상세보기 페이지(P5)

내부기능

1. 호선 탭 생성
 - 노선도 페이지에서 보낸 intent 데이터 curStation에 저장
 - DB subway_info의 subway_line 테이블 참조하는 cursor_line 생성
 (where NAME = curStation)
 - while(cursor_line.moveToNext)를 사용하여 커서의 모든 값을
 배열 Staionline[]에 저장 (해당 역의 호선 값)

 1.1. 탭 생성
 - addTab사용하여 모든 호선에 해당하는 탭을 추가한다.
 - 모든 탭을 GONE처리 하여 보이지 않고, 공간도 차지않게 설정한다.
 - Staionline[]에 저장된 값에 해당하는 호선의 탭을 활성화한다.
 - 위에서 탭을 활성화 할 때 Bundle값으로 curStation을 호선fragment에 보낸다.

 1.1.1. 탭의 첫 번째 화면 설정 및 탭 클릭 시 화면전환
 - Staionline의 첫 번째 값으로 탭의 첫 번째 화면을 fragment로 설정한다.
 - 클릭된 탭의 position값에 따라 fragment를 전환한다.

2. 호선 정보 프라그먼트 생성

 2.1 현재 역, 이전 역, 다음 역 텍스트 설정
 - 위에서 보낸 Bundle값 curStation을 curStation_b에 저장한다.
 - curStation_b를 현재 역 텍스트에 설정한다.
 - DB subway_info의 subway_line 테이블 CODE값 참조하는 cursor_code 생성
 (where NAME = curStation_b and line=(현재 fragment 호선))
 - cur_code에 커서 값을 저장하고, nextCode에는 cur_code-1,
 beforeCode에는 cur_code+1을 저장한다.
 - 커서로 subway_line 테이블의 NAME값을 참조한다.
 (where CODE = nextCode/beforeCode)
 - 커서에 저장된 NAME값을 각각 다음 역, 이전 역 텍스트에 설정한다.

역 상세보기 페이지(P5)

2.2. 다음 역, 이전 역 버튼 클릭
- 다음 역 버튼 클릭 시 cur_code = cur_code-1, nextCode에는 cur_code-1, beforeCode에는 cur_code+1을 저장한다.
- 커서로 subway_line 테이블의 NAME값을 참조한다.
(where CODE = curCode/nextCode/beforeCode)
- 커서에 저장된 NAME값을 각각 현재 역, 다음 역, 이전 역 텍스트에 설정한다.
- 도착정보를 갱신한다.
- 다음 역 버튼 클릭 시 cur_code = cur_code+1해주는 것 이외에는 동일하다.

2.3. 도착정보 세팅

2.3.1. 인터넷 체크
- ConnectivityManager 및 NetworkInfo를 사용하여 isConnect에 인터넷 유무 저장

2.3.2 인터넷이 있을 때 도착정보 세팅 - 크롤링
- Bundle값 curStation을 다시 불러와서 curStation_b에 저장한다.
- Thread를 생성하고 try-catch문으로 작성한다.
- URL에 크롤링할 주소를 적는다.(링크+curStation_b+(현재 fragment 호선))
- Document객체 doc에 Jsoup으로 링크를 읽은 값을 저장한다.
- Elements에 doc에서 긁어올 부분의 cssSelecor값을 입력하여 가져온다.
- □행과 □분에 해당하는 부분을 파싱하여 도착정보에 세팅한다.

2.3.3. 인터넷이 없을 때 도착정보 세팅
- Bundle값 curStation을 다시 불러와서 curStation_b에 저장한다.
(이 때 curStation_b = curStation + (현재 fragment 호선))
- SimpleDateFormat을 사용하여 현재 시간, 분을 저장한다.
- DB subway_schedule의 schedule 테이블을 참조하는 UP_cursor 생성
(where NAME = curStation_b)
- 커서 데이터(시간표 데이터)에서 텍스트 부분과 숫자 부분을 차례로 배열에 저장
(replaceAll으로 공백처리 및 공백 기준으로 값 저장)
- 위에서 저장시킨 배열의 길이만큼 반복하여 (반복수 i)
만약 현재시간보다 시간표에 저장된 시간이 크다면 i값에 해당하는 텍스트와 분 세팅
- 나머지도 같은 방식으로 세팅한다.

외부기능

1. 경로보기 버튼
- intent로 curStation(현재역) 데이터를 경로 보기 페이지 (P6)로 보내고 이동한다.

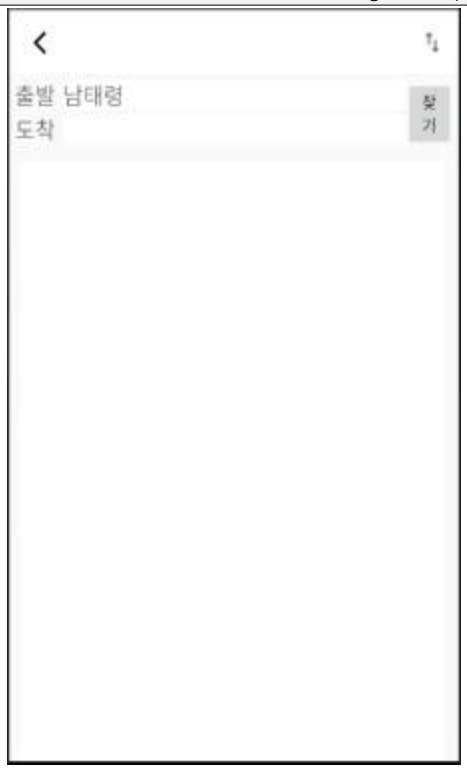
2. 시간표 버튼
- intent로 curStation(현재역) 데이터를 시간표 페이지 (P7)로 보내고 이동한다.

3. 역 내부 안내 버튼
- intent로 curStation(현재역) 데이터를 역 내부 안내 페이지(P8)로 보내고 이동한다.

4. 문의하기 버튼
- 클릭 시 intent로 문의 페이지 띄우기(P9)

5. 뒤로가기 버튼
- 클릭 시 finish()를 이용하여 이전 화면으로 이동

경로보기 페이지(P6)



<검색 전>

<검색 후>

내부기능

1. 지하철 경로 정보 세팅

1.1. 출발/도착역 입력 후 찾기 버튼 클릭 시 정보 제공

- 출발/도착역 텍스트를 start, arrival에 각각 저장
- DB subway_info의 subway_coordinate 테이블을 참조하는 route_cusor 생성
- while(route_cusor .moveToNext)를 사용하여 커서에 저장된 코드값 저장
- 코드값을 알고리즘의 노드값과 대응되게 조정하고 알고리즘 시작
  (반환 값 : 소요시간, 배열 route[]에 경로 저장)
- 소요시간은 route_time에 저장
- route[]에 저장된 경로노드 번호를 CODE값에 맞게 조정
- route_cusor를 moveToFirst하여 첫 번째 위치로 바꾸고
  while (route_name.moveToNext) 사용하여
  CODE값과 일치하는 역 이름을 배열 subway_route에 저장(호선도 배열에 저장)
- fragment에 역 이름/호선 배열, 소요시간 정보 번들 데이터로 보냄

1.2.. fragment 생성 - 소요시간 및 정거장 정보

- 위에서 받은 소요시간 정보 번들 값을 소요시간 텍스트에 설정한다.
- 역 이름 배열을 탐색하여 n번쨔와 n+1번째 이름이 같으면 transfer_Count++
- transfer_Count값에 따라 경로 UI fragment를 생성하고
  위에서 받은 역 이름/호선 배열 번들 데이터를 fragmetn로 다시 보냄

경로보기 페이지(P6)
1.3. fragemnt 생성 - 경로 UI 생성 1.3.1. 환승역이 0개 일 때 - 위에서 보낸 역 이름/호선 배열 번들 데이터를 배열 route와 route_line에 저장 - 호선 이미지와 막대 이미지의 색을 swich문에서 route_line[0]값 기준으로 설정한다. - 텍스트를 설정한다. 1.3.2. 환승역이 1개 일 때 - 위에서 보낸 역 이름/호선 배열 번들 데이터를 배열 route와 route_line에 저장 - route를 탐색하여 n번 값과 n+1값이 같다면 배열 transfer에 n을 저장한다. - transfer_count에 값에 따라 텍스트를 설정하고 transfer_count++을 해준다. - 이미지 설정은 transfer값들 기준으로 순차적으로 설정한다. 2. 스왑 버튼 - 클릭 시 출발 역과 도착 역의 텍스트가 변환된다. - temp에 출발 역 텍스트를 저장한다 - 출발 역 텍스트에 도착 역 텍스트를 설정한다. - 도착 역 텍스트에 temp에 저장한 텍스트를 설정한다. **외부기능** 1. 뒤로가기 버튼 - 클릭 시 finish()를 이용하여 이전 화면으로 이동

시간표 페이지(P7)

내부기능

1. 시간표 정보 세팅

1.1. 현재 역, 라디오 버튼 생성

- curStation(현재역) intent 데이터를 받아서 curStation에 저장한다.
- curStation+호선을 현재 역 텍스트에 설정한다.
- 라디오 버튼을 초기화하고, 클릭 시 swictc문으로 해당하는 fragment로 전환.

1.2. 방향 텍스트 설정

- DB subway_schedule의 schedule 테이블을 참조하는 DIR_cursor 생성 (where NAME = curStation)
- DIR_cursor의 0번 값은 상행, 1번 값은 하행 텍스트로 설정한다.

1.3. 시간표 텍스트 설정 - 상행

- RecyclerView를 생성한다.
- DB subway_schedule의 schedule 테이블을 참조하는 UP_cursor 생성 (where NAME = curStation and TYPE="상행")
- 커서 데이터(시간표 데이터)에서 텍스트 부분과 숫자 부분을 차례로 텍스트에 저장 (replaceAll으로 공백처리 및 공백 기준으로 값 저장)
- 생성된 텍스트를 RecyclerView에 ScheduleAdapter사용하여 additem (05~24시 반복)

1.3.1. 현재 시간대로 스크롤

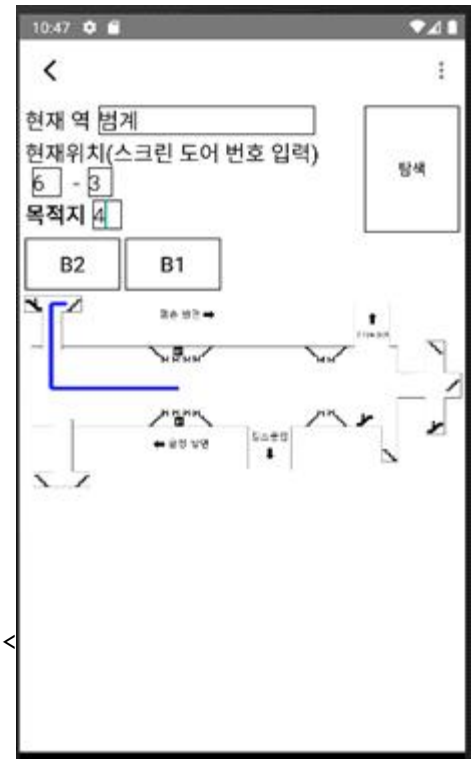
- SimpleDateFormat을 이용하여 현재시간을 구하여 저장한다.
- scrollToPositionWithOffset을 이용하여 시간대 별로 해당 위치로 이동한다.

외부 기능

1. 뒤로가기 버튼

- 클릭 시 finish()를 이용하여 이전 화면으로 이동

역 내부 안내 페이지(P8)



내부기능

1. 역 내부 경로 찾기

1.1. 사용자 입력(다음 버튼 클릭 시 step변수의 값에 따라 실행한다. step 초기값 = 0)
※아래의 이미지는 DB에 저장된 bitmap을 Canvas로 설정하여 ImageView에 세팅

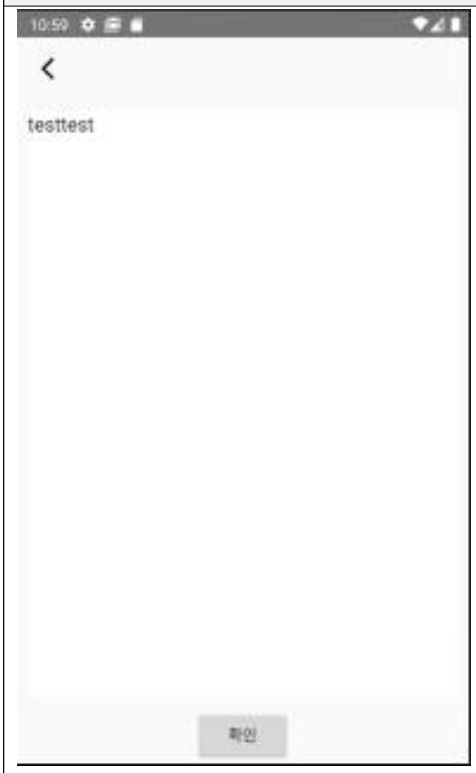
1.1.1 step==0 : 외부 기능1번으로 역 이름이 입력 후 다음 버튼
- 입력이 됐다면 출발지점이 있는 층의 이미지를 세팅(지하철에서 내리는 층), step=1
- 입력이 되지 않았다면 입력이 되지 않았다는 토스트 메시지 출력

1.1.2 step==1 : 출발지점을 입력 후 다음 버튼
- 입력이 됐다면 도착지점이 있는 층의 이미지를 세팅(출구 층),step=2
- 입력이 되지 않았다면 입력이 되지 않았다는 토스트 메시지 출력

1.1.3 step==2 : 도착지점 입력 후 다음 버튼
- 입력이 됐다면 경로를 알려주는 선을 그리고, 각 층의 이미지를 볼 수 있는 버튼을 생성한다.
- 입력이 되지 않았다면 입력이 되지 않았다는 토스트 메시지 출력

역 내부 안내 페이지(P8)
2. 경로 안내 그림 2.1. 그래프 생성 및 탐색 - 입력된 역에 해당하는 DB를 찾아서 각 노드번호, 거리 값을 그래프에 넣는다. - 입력된 출발지점과 도착지점의 최단거리를 찾는 다익스트라 알고리즘 실행 후 경로 배열을 반환 받는다. 2.2. 경로 안내 선 DB에 저장된 좌표값 기준으로 캔버스에 DrawLine()함수로 선을 그린다. (DB에 시작A,도착B....으로 저장되어 있으므로 반환된 경로 배열 n,n+1값이 A,B값과 동일하면 그림) 각 화면에 대한 경로를 보여줄때 경로 배열에서 각 층에 해당하는 값만을 매개변수로 보낸다. **외부기능** 1. 역 내부 경로 찾기 1.1. 사용자 입력(다음 버튼 클릭 시 step변수의 값에 따라 실행한다. step 초기값 = 0) ※아래의 이미지는 DB에 저장된 bitmap을 Canvas로 설정하여 ImageView에 세팅 1.1.1 step==0 : 외부 기능1번으로 역 이름이 입력 후 다음 버튼 - 입력이 됐다면 출발지점이 있는 층의 이미지를 세팅(지하철에서 내리는 층), step=1 - 입력이 되지 않았다면 입력이 되지 않았다는 토스트 메시지 출력 1.1.2 step==1 : 출발지점을 입력 후 다음 버튼 - 입력이 됐다면 도착지점이 있는 층의 이미지를 세팅(출구 층),step=2 - 입력이 되지 않았다면 입력이 되지 않았다는 토스트 메시지 출력 1.1.3 step==2 : 도착지점 입력 후 다음 버튼 - 입력이 됐다면 경로를 알려주는 선을 그리고, 각 층 이미지를 볼 수 있는 버튼 생성 - 입력이 되지 않았다면 입력이 되지 않았다는 토스트 메시지 출력 2. 경로 안내 그림 2.1. 그래프 생성 및 탐색 - 입력된 역에 해당하는 DB를 찾아서 각 노드번호, 거리 값을 그래프에 넣는다. - 입력된 출발지점과 도착지점의 최단거리를 찾는 다익스트라 알고리즘 실행 후 경로 배열을 반환 받는다. 2.2. 경로 안내 선 DB에 저장된 좌표값 기준으로 캔버스에 DrawLine()함수로 선을 그린다. (DB에 시작A,도착B....으로 저장되어 있으므로 반환된 경로 배열 n,n+1값이 A,B값과 동일하면 그림) 각 화면에 대한 경로를 보여줄때 경로 배열에서 각 층에 해당하는 값만을 매개변수로 보낸다. 외부기능 1. 역 이름 작성 부분(EditText) - 클릭 시 역 검색 페이지(P4)로 이동 2. 뒤로가기 버튼 - 클릭 시 fnish() 이용하여 이전화면으로 이동

문의 페이지(P9)



내부기능

1. 문의
 1.1. 이용자가 문의한 내용을 받을 이메일을 설정
  - address라는 String 배열에 값 저장
 1.2. 이메일 보내기
  - address와 제목, 본문 내용을 putExtra로 데이터를 보냄
 1.3 문의 페이지 닫기
  - finish()를 이용하여 이메일 보낸 후 이전 화면으로 돌아가게 설정

외부기능

1. 뒤로가기 버튼
 - 클릭 시 finish()를 이용하여 이전 화면으로 이동

설정 페이지(P10)

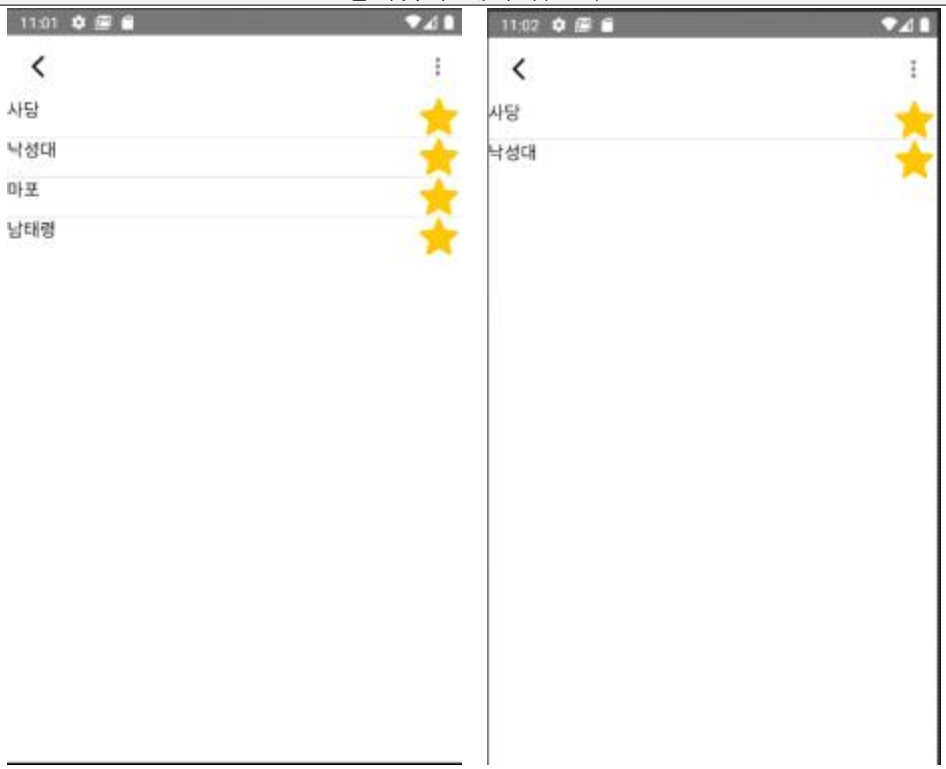
내부기능

1. 설정 페이지 구성하기
 1.1. 설정 화면 구성할 리스트 생성
  - 리스트배열 생성 후 즐겨찾기, 문의하기 값을 넣음
 1.2. 화면에 보이기
  - setAdapter를 이용하여 생성했던 리스트를 화면에 보이게 함.

외부기능

1. 즐겨찾기 버튼
 - 즐겨찾기 클릭 시 intent로 즐겨찾기 페이지(P11) 띄우기
2. 문의하기 버튼
 - 문의하기 클릭 시 intent로 문의하기 페이지(P9) 띄우기
3. 뒤로가기 버튼
 - 뒤로가기 버튼 클릭 시 finish()를 이용하여 이전 화면으로 이동

즐겨찾기 페이지(P11)



내부기능

1. 즐겨찾기 리스트 보여주기
 1.1. 리스트뷰 커스텀
  - converview를 이용하여 텍스트와 이미지버튼(노란 별)으로 커스텀
 1.2. 리스트 채우기
  - subway_bookmark 테이블에서 데이터 조회 후 모든 데이터를 리스트에 추가 한다.

2. 즐겨찾기 해제
  - 리스트에서 이미지버튼을 클릭 시 해당 역의 문자열을 subway_bookmaer에서 데이터 삭제 후 Search_History에 데이터 추가

외부기능

1. 뒤로가기 버튼
 - 클릭 시 finish()를 이용하여 이전 화면으로 이동