소프트웨어 설계서

작성자: 임종원, 박태형, 이동현, 김민수, 조민선

**변경 이력**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 버전 | 변경 내역 | 작성 | | 검토 | | 승인 | |
| 일자 | 담당자 | 일자 | 담당자 | 일자 | 담당자 |
| 1.0.1 | 클래스, 유스케이스 다이어그램 작성 | 11/02 | 김민수 | 11/02 | 임종원 | 11/02 | 조민선 |
| 1.0.2 | 상태 다이어그램 작성 | 11/03 | 박태형 | 11/03 | 이동현 | 11/03 | 조민선 |
| 1.0.3 | MVC패턴 작성 | 11/05 | 박태형 | 11/05 | 이동현 | 11/05 | 조민선 |
| 1.0.4 | 클래스 다이어그램 수정 | 11/05 | 김민수 | 11/05 | 임종원 | 11/05 | 조민선 |
| 1.0.5 | 액티비티 다이어그램 작성 | 11/05 | 이동현 | 11/05 | 박태형 | 11/05 | 조민선 |
| 1.0.6 | 시퀀스 다이어그램 작성 | 11/07 | 임종원 | 11/07 | 김민수 | 11/07 | 조민선 |
| 1.0.7 | 액티비티 다이어그램 수정 | 11/09 | 이동현 | 11/09 | 박태형 | 11/09 | 조민선 |
| 1.0.8 | 시퀀스 다이어그램 수정 | 11/09 | 임종원 | 11/09 | 김민수 | 11/09 | 조민선 |

## 문서 개요

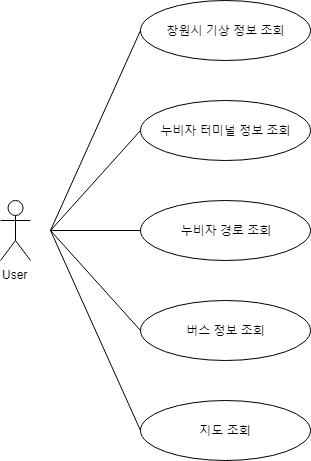
본 문서는 요구사항 명세서를 바탕으로 명시된 요구사항을 만족시키는 방안을 구체화한 결과로서 소프트웨어 설계를 기술한다.

본 문서의 주요 구성 항목은 다음과 같다.

* UML Diagram: 소프트웨어의 구성을 use case, activity, class, sequence, state의 5가지 다이어그램으로 정의한다.
* Software architecture: 소프트웨어 구조를 정의한다.

## UML Diagram

1. **Use case diagram**



* 1. **뚜비: 창원시 기상 정보 조회**

|  |  |
| --- | --- |
| **Actor** | User, 창원시 누비자-버스 안내 시스템(뚜비) |
| **Description** | 유저는 기상 정보를 제공받을 수 있다. |
| **Data** | 창원시 기상 정보 |
| **Stimulus** | 유저의 기상 정보 요청 이벤트 발생 |
| **Response** | 기상 정보가 수신되었음을 확인한다. |
| **Comments** | 유저는 인터넷이 연결된 디바이스를 사용하여야 한다. |

* 1. **뚜비: 누비자 터미널 조회**

|  |  |
| --- | --- |
| **Actor** | User, 창원시 누비자-버스 안내 시스템(뚜비) |
| **Description** | 유저는 누비자 터미널 정보를 제공받을 수 있다. |
| **Data** | 누비자 터미널 정보 |
| **Stimulus** | 유저의 누비자 터미널 정보 요청 이벤트 발생 |
| **Response** | 누비자 터미널 정보가 수신되었음을 확인한다. |
| **Comments** | 유저는 인터넷이 연결된 디바이스를 사용하여야 한다. |

* 1. **뚜비: 누비자 경로 조회**

|  |  |
| --- | --- |
| **Actor** | User, 창원시 누비자-버스 안내 시스템(뚜비) |
| **Description** | 유저는 누비자 경로를 제공받을 수 있다. |
| **Data** | 누비자 경로 정보 |
| **Stimulus** | 유저의 누비자 경로 정보 요청 이벤트 발생 |
| **Response** | 누비자 경로 정보가 수신되었음을 확인한다. |
| **Comments** | 유저는 인터넷이 연결된 디바이스를 사용하여야 한다. |

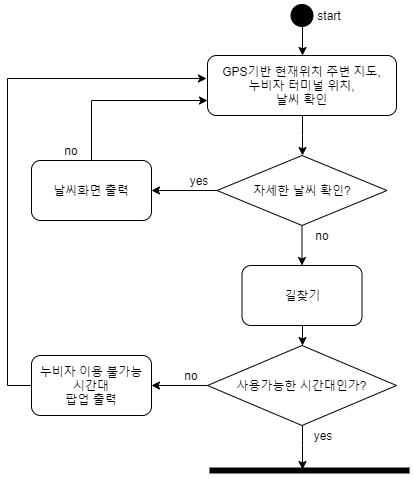
* 1. **뚜비: 버스 정보 조회**

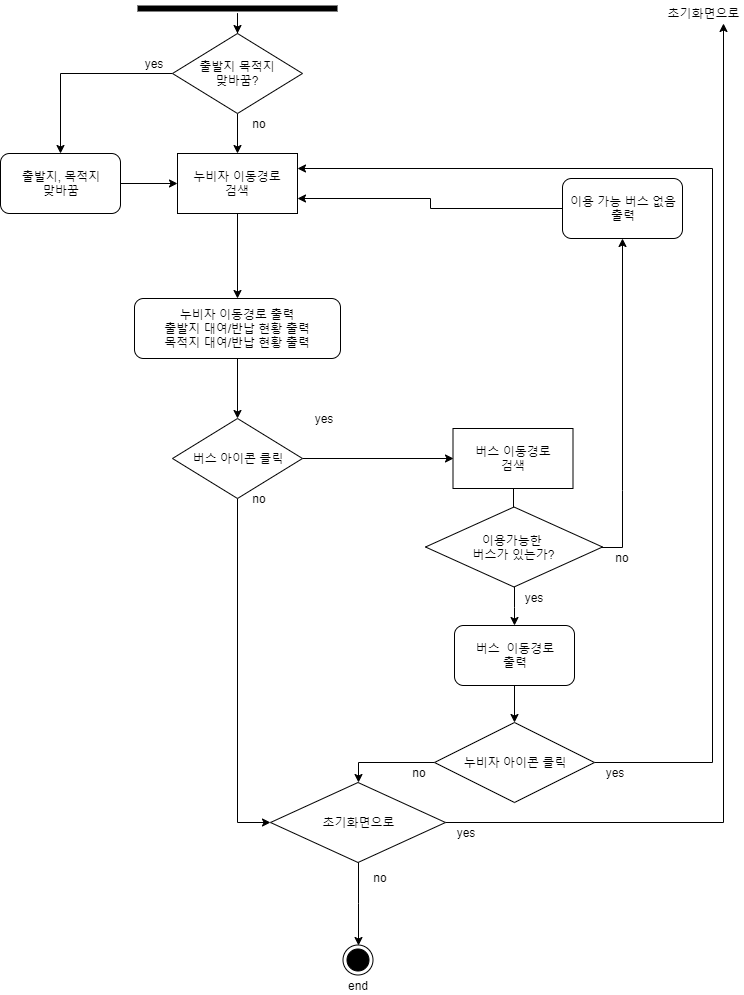
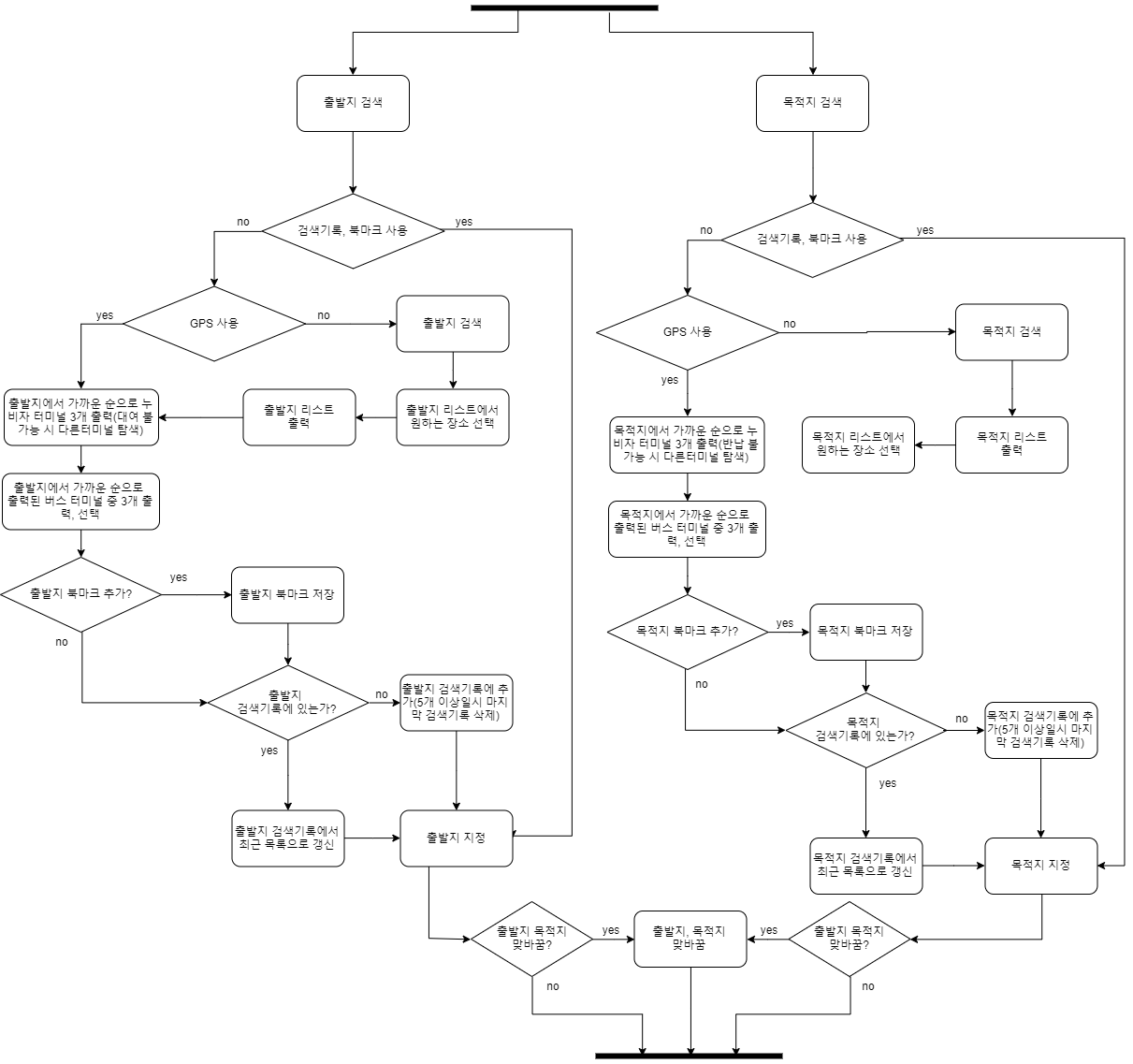
|  |  |
| --- | --- |
| **Actor** | User, 창원시 누비자-버스 안내 시스템(뚜비) |
| **Description** | 유저는 버스 정보를 제공받을 수 있다. |
| **Data** | 버스 정보 |
| **Stimulus** | 유저의 버스 정보 요청 이벤트 발생 |
| **Response** | 버스 정보가 수신되었음을 확인한다. |
| **Comments** | 유저는 인터넷이 연결된 디바이스를 사용하여야 한다. |

* 1. **뚜비: 지도 조회**

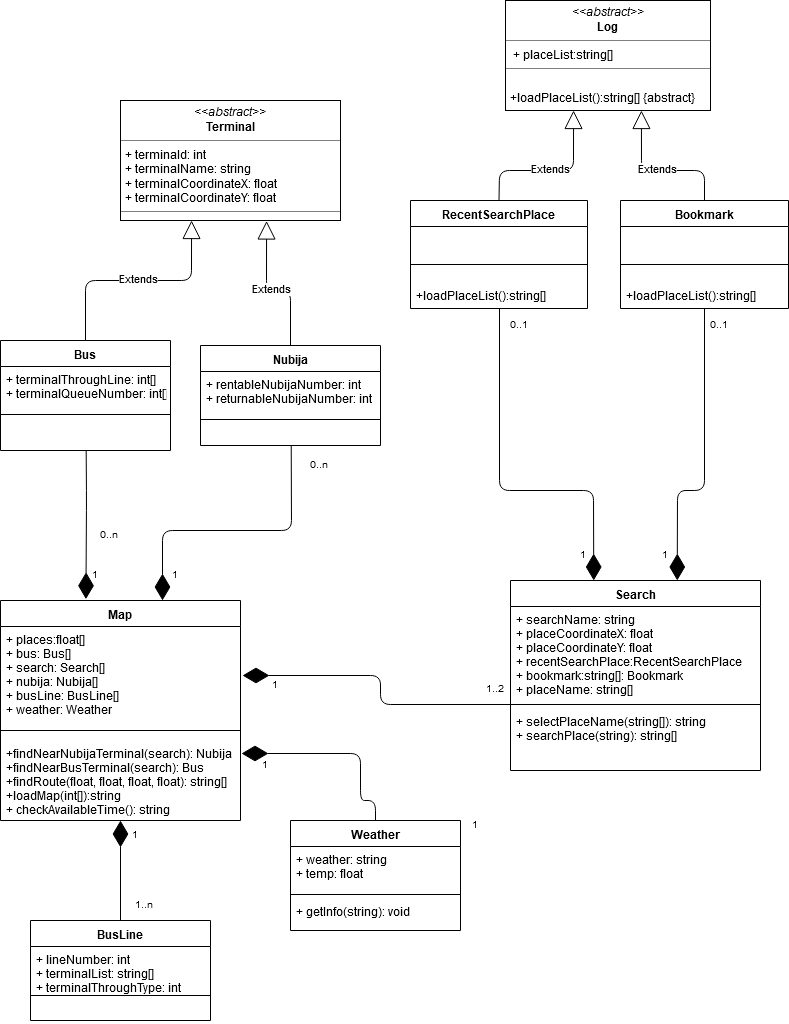
|  |  |
| --- | --- |
| **Actor** | User, 창원시 누비자-버스 안내 시스템(뚜비) |
| **Description** | 유저는 지도를 제공받을 수 있다. |
| **Data** | 지도 |
| **Stimulus** | 유저의 지도 요청 이벤트 발생 |
| **Response** | 지도가 수신되었음을 확인한다. |
| **Comments** | 유저는 인터넷이 연결되고 GPS가 사용 가능한 디바이스를 사용하여야 한다. |

1. **Activity diagram**





1. **Class diagram**



* 1. **뚜비: Terminal class**

|  |  |
| --- | --- |
| 뚜비: Terminal class | |
| 설명 | 교통수단의 터미널에 관한 추상 클래스 |
| 필드 - terminalId | 해당 터미널의 고유번호 |
| 필드 - terminalName | 해당 터미널의 명칭 |
| 필드 - terminalCoordinateX | 해당 터미널의 위도 |
| 필드 - terminalCoordinateY | 해당 터미널의 경도 |

* 1. **뚜비: Bus class**

|  |  |
| --- | --- |
| 뚜비: Bus class | |
| 설명 | 버스 터미널의 정보에 대한 클래스 - Terminal 클래스를 상속한다 |
| 필드 - terminalThroughLine | 해당 터미널에 정차하는 노선들의 번호 |
| 필드 - terminalQueueNumber | 노선별 해당 정류장의 정차순서 |

* 1. **뚜비: Nubija class**

|  |  |
| --- | --- |
| 뚜비: Nubija class | |
| 설명 | 누비자 터미널의 정보에 대한 클래스 - Terminal 클래스를 상속한다 |
| 필드 - rentableNubijaNumber | 해당 터미널에서 대여 가능한 누비자의 댓수 |
| 필드 - returnableNubijaNumber | 해당 터미널에 반납 가능한 누비자의 댓수 |

* 1. **뚜비: Weather class**

|  |  |
| --- | --- |
| 뚜비: Weather class | |
| 설명 | 창원지역의 기상정보에 대한 클래스 |
| 필드 - weather | 창원지역의 날씨 |
| 필드 - temp | 창원지역의 온도 |
| 메소드 - getInfo | 기상정보를 외부 시스템에서 가져오는 클래스 |

* 1. **뚜비: Log class**

|  |  |
| --- | --- |
| 뚜비: Log class | |
| 설명 | 저장된 위치에 대한 추상 클래스 |
| 필드 - placeList | 저장된 위치 |
| 메소드 - loadPlaceList() | 위치정보 파일을 불러오는 추상 메소드 |

* 1. **뚜비: RecentSearchPlace class**

|  |  |
| --- | --- |
| 뚜비: RecentSearchPlace class | |
| 설명 | 최근 사용된 위치에 대한 클래스 - Log클래스를 상속한다 |
| 메소드 - loadPlaceList() | 최근 사용된 위치가 저장된 파일을 불러온다 |

* 1. **뚜비: Bookmark class**

|  |  |
| --- | --- |
| 뚜비: Bookmark class | |
| 설명 | 북마크로 저장한 위치에 대한 클래스 - Log클래스를 상속한다 |
| 메소드 - loadPlaceList() | 북마크로 저장한 위치가 저장된 파일을 불러온다 |

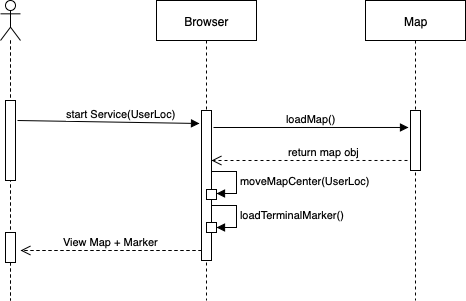
* 1. **뚜비: Search class**

|  |  |
| --- | --- |
| 뚜비: Search class | |
| 설명 | 위치 검색에 대한 클래스 |
| 필드 - searchName | 유저가 입력한 명칭 |
| 필드 - placeCoordinateX | 지정된 위치의 위도 |
| 필드 - placeCoordinateY | 지정된 위치의 경도 |
| 필드 - placeName | 유저가 입력한 명칭으로 검색된 장소들의 목록 |
| 필드 - recentSearchPlace | RecentSearchPlace 클래스로 생성된 객체 |
| 필드 - bookmark | Bookmark 클래스로 생성된 객체 |
| 메소드 - selectPlacePlaceName() | 여러 위치들중 하나를 지정한다 |
| 메소드 - searchPlace() | 유저가 입력한 명칭으로 위치를 검색한다 |

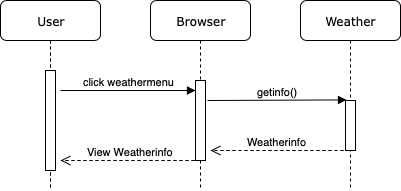
* 1. **뚜비: Busline class**

|  |  |
| --- | --- |
| 뚜비: BusLine class | |
| 설명 | 버스 노선에 대한 클래스 |
| 필드 - lineNumber | 버스 노선의 고유번호 |
| 필드 - terminalList | 해당 노선이 정차하는 정류장 |
| 필드 - terminalThroughType | 해당 노선의 상/하행 정보 |

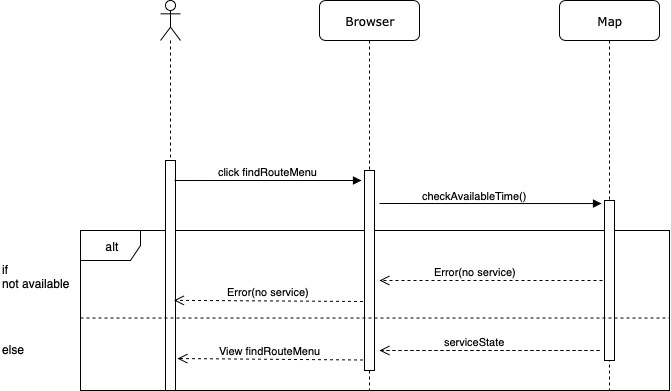
1. **sequence diagram**
   1. **메인 화면**



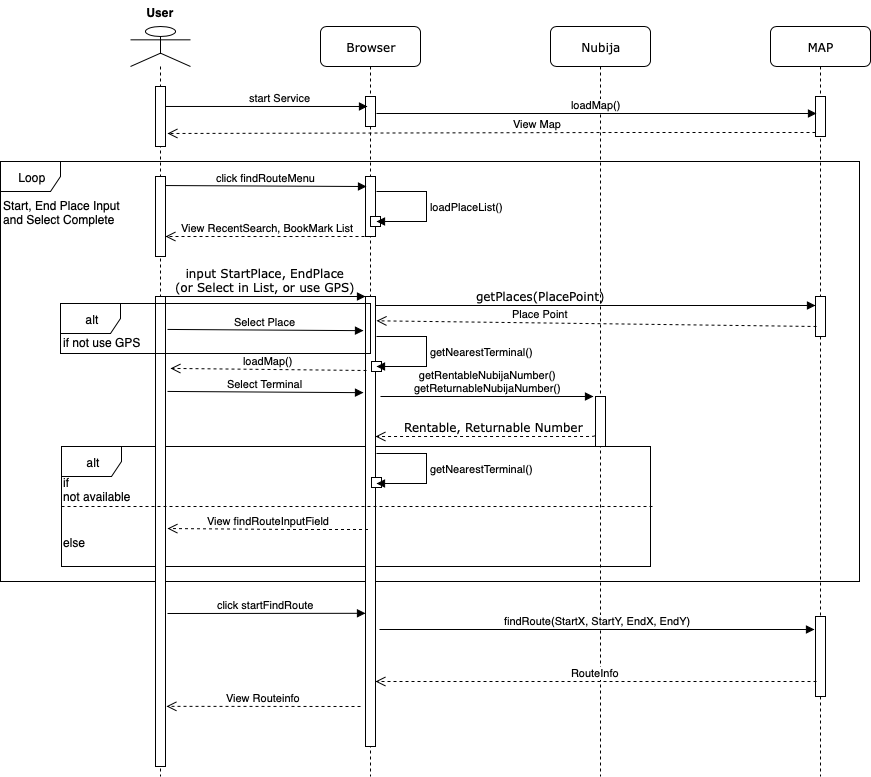
* 1. **기상 정보**



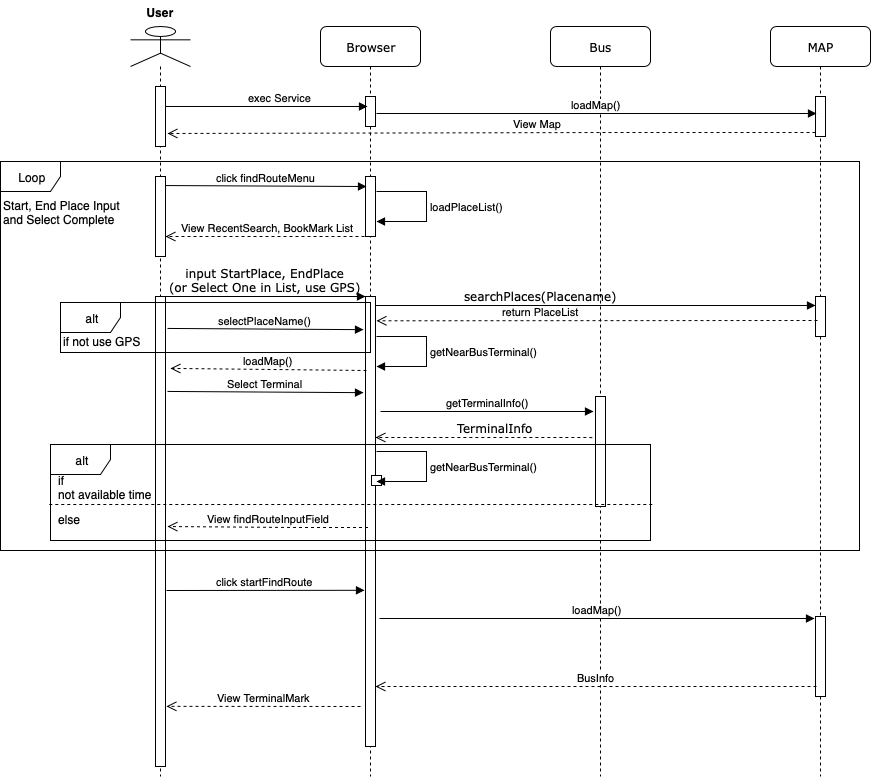
* 1. **서비스 시간 검사**



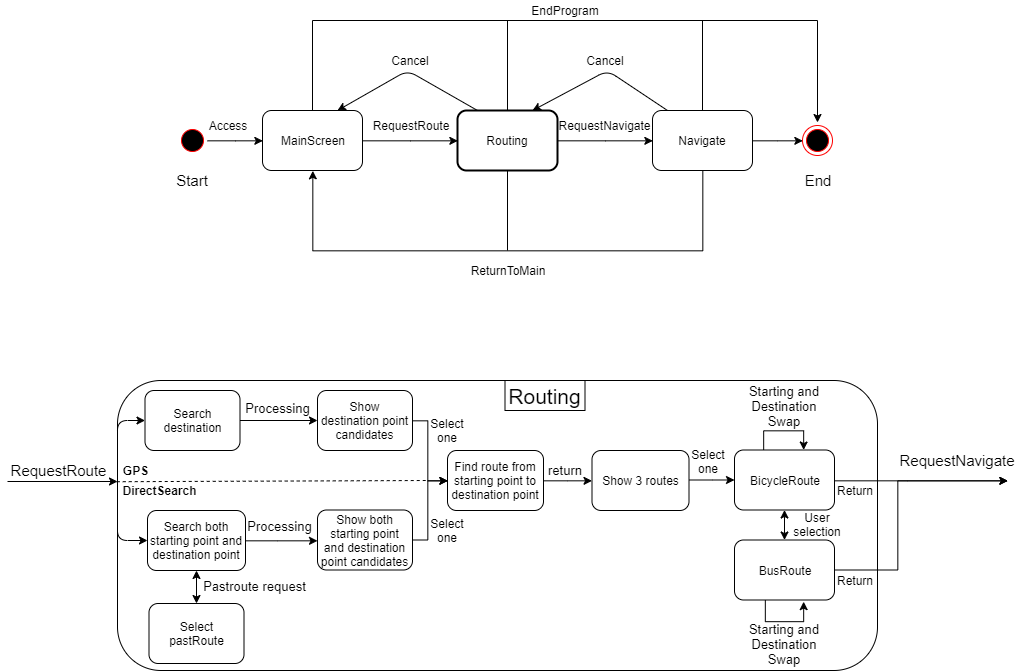
* 1. **누비자 경로 검색**



* 1. **버스 노선 검색**



1. **State diagram**



* 1. **Overview**

|  |  |
| --- | --- |
| **State** | **Description** |
| MainScreen | Main page shows weather information, a map includes user’s location, and search bar for find routes. |
| Routing | Web page in routing. User can select routes from stating point to destination point. Each starting point can be user’s present position or selected position by user. |
| Navigate | Navigate page shows a route selected by user. |

* 1. **Routing**

|  |  |
| --- | --- |
| **State** | **Description** |
| Search destination | Search destination. |
| Show destination point candidates | Show destination candidates and wait for user’s selection. |
| Search both starting point and destination point | Wait for user’s inputs for selecting route. |
| Select pastRoute | Select past route as starting point or destination point |
| Show both Starting point and destination point candidates | Show starting point and destination point candidates and wait for user’s selection. |
| Find route from starting point to destination point | Get user’s input and set route. |
| Show 3 routes | Wait for user’s selection. 3 routes include nearest stations from starting point. |
| bicycleRoute | Show route from starting point to destination point by bicycle. Starting and destination point can be swapped by user. |
| busRoute | Show route from starting point to destination point by bus. Starting and destination point can be swapped by user. |

## Software architecture

* MVC pattern

