



와따렌트 포팅 매뉴얼

1 프로젝트 기술 스택

- A. FE
- B. EMB
- C. 기타

2 빌드 방법

- A. 프론트엔드 빌드 방법

3 EMB 매뉴얼

- A. Morai Simulator 설치
- B. VirtualBox 설치
- C. Ubuntu 20.04 설치
- D. ROS 설치
- E. 실행

4 데이터 통신

- A. FE to DB
- B. EMB to DB
- C. DB to FE
- D. DB to EMB

5 시나리오

- A. 홈 화면
- B. 메인 화면
- C. 차량 픽업 화면
- D. 목적지 설정 화면

1 프로젝트 기술 스택

A. FE

- 기술 스택 (버전)
 - React.js 18
- 사용 툴 (버전)
 - Visual Studio Code 1.74.2

B. EMB

- 기술 스택 (버전)
 - Python3 3.6.9
 - Morai Simulator 22.R4.1
 - ROS 1.14.13
- 사용 툴 (버전)
 - Oracle VM VirtualBox 7.0.6
 - Visual Studio Code 1.77.0

C. 기타

- 형상 관리: GitLab
- 이슈 관리: Jira

- 커뮤니케이션: Notion, Mattermost
- 디자인: Figma, Wondershare Filmora, PowerPoint 2016
- 데이터베이스: Firebase Realtime DB
- 배포: Firebase Hosting

2 빌드 방법

A. 프론트엔드 빌드 방법

1. `npm install -g firebase-tools`
2. `firebase login`

```
C:\Users\king1\OneDrive\바탕 화면\emotion-diary>firebase login
i Firebase optionally collects CLI usage and error reporting information to help improve our
products. Data is collected in accordance with Google's privacy policy (https://policies.google.com/privacy) and is not used to identify you.

? Allow Firebase to collect CLI usage and error reporting information? (Y/n) █
```

3. y입력 후 엔터
4. 구글 로그인창 오픈 후 해당되는 아이디 클릭후 권한 허용

```
Waiting for authentication...

+ Success! Logged in as ina9377@gmail.com
PS D:\workspace\workspace_oneBiteReact\emotiondiary> █
```

5. 로그인 완료
6. `firebase init`

```
PS D:\workspace\workspace_oneBiteReact\emotiondiary> firebase init

##### 
##      ##      ##      ##      ##      ##      ##      ##      ##      ##
##### 
##      ##      ##      ##      ##      ##      ##      ##      ##      ##
##      ##      ##      ##      ##      ##      ##      ##      ##      ##
##      ##      ##      ##      ##      ##      ##      ##      ##      ##

You're about to initialize a Firebase project in this directory:

D:\workspace\workspace_oneBiteReact\emotiondiary
```

```
? Are you ready to proceed? Yes
? Which Firebase features do you want to set up
to select features, then Enter to confirm your
```

```

no (optionally) provision default instance
( ) Firestore: Configure security rules and indexes files for Firestore
( ) Functions: Configure a Cloud Functions directory and its files
>(*) Hosting: Configure files for Firebase Hosting and (optionally) set up GitHub Action deploys
( ) Hosting: Set up GitHub Action deploys
( ) Storage: Configure a security rules file for Cloud Storage

```

```

? Please select an option: (Use arrow keys)
> Use an existing project
  Create a new project
  Add Firebase to an existing Google Cloud Platform project
  Don't set up a default project

```

```

? Select a default Firebase project for this directory: (Use arrow keys)
> react-diary-project-fc482 (react-diary-project)

```

!! 중요 !!

```

? What do you want to use as your public directory? build

```

Woohoo!

Firebase CLI GitHub Login
Successful

You are logged into GitHub via the Firebase CLI. You can immediately close this window and continue using the Firebase CLI.

7. `firebase deploy`

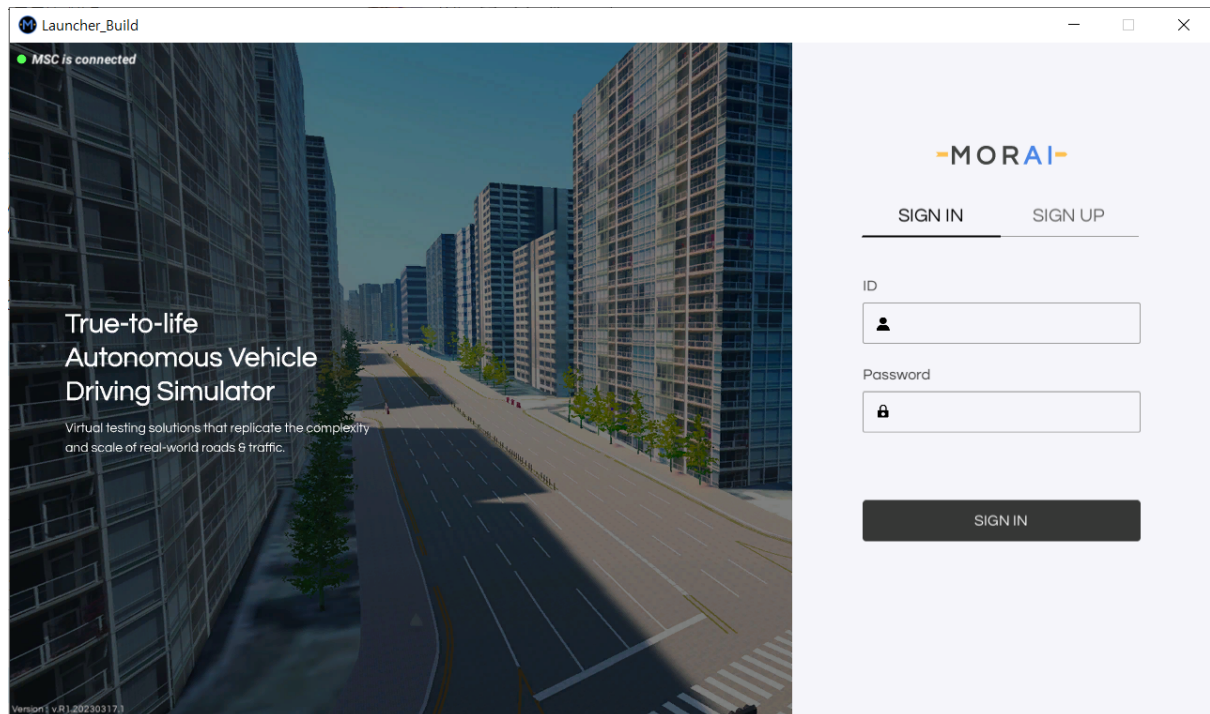
8. docker 컨테이너 생성

`doc`

3 EMB 매뉴얼

A. Morai Simulator 설치

MoraiLauncher_Win.zip 압축 해제 후 MoraiLancher_Win.exe 파일 실행



B. VirtualBox 설치

<https://www.virtualbox.org/wiki/Downloads> 에서 최신 버전 Windows hosts 설치



C. Ubuntu 20.04 설치

1. <https://releases.ubuntu.com/20.04/> 에서 64-bit PC (AMD64) desktop imgae 다운로드

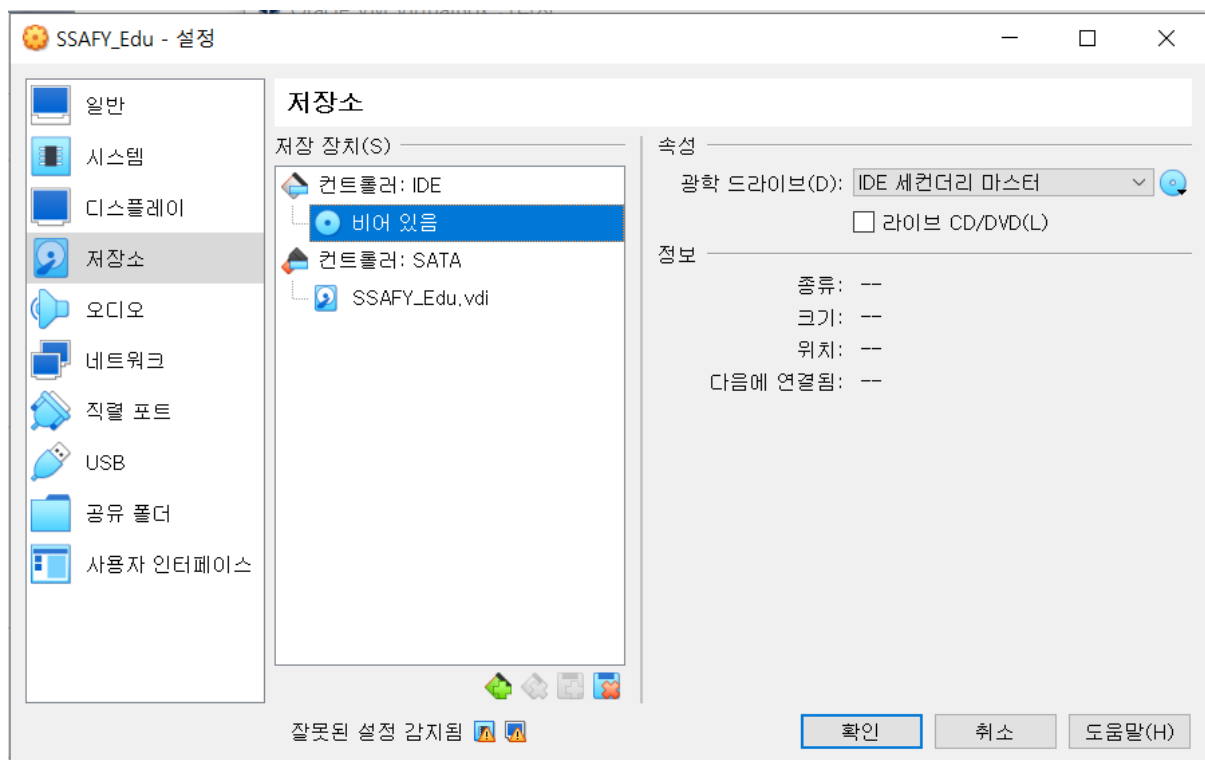
Desktop image

The desktop image allows you to try Ubuntu without changing your computer at all, and at your option to install it permanently later. This type of image is what most people will want to use. You will need at least 1024MiB of RAM to install from this image.

64-bit PC (AMD64) desktop image

Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

2. VMWare > 설정 > 저장소 > 컨트롤러 IDE > 광학 드라이브에 Ubuntu 20.04 iso 설치



D. ROS 설치

1-1. ROS 기본 설정

```
$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
$ sudo apt install curl
$ curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -
$ sudo apt update
$ sudo apt install ros-melodic-desktop-full
$ sudo apt-get install python-rosdep
$ sudo rosdep init
$ rosdep update
$ echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
```

```
$ source ~/.bashrc
$ sudo apt install python-rosinstall python-rosinstall-generator pythonwstool build-essential
```

1-2. ROS 설치 확인

```
$ roscore
```

1-3. Morai msg 파일 다운

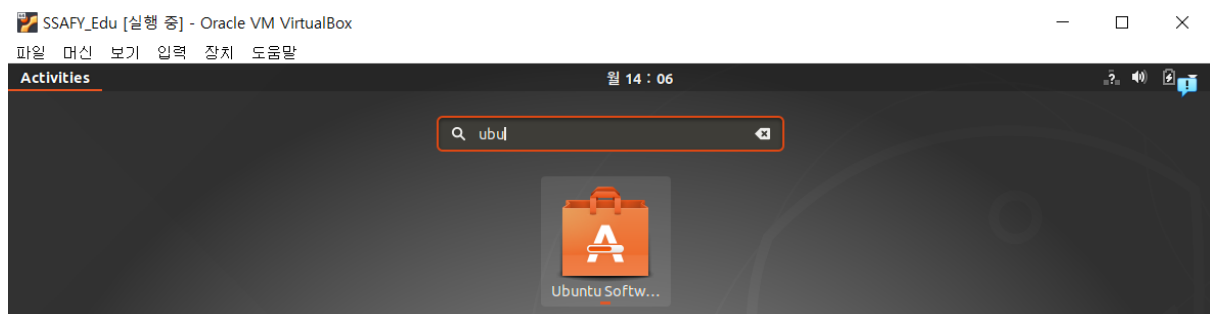
```
$ cd ~/catkin_ws
$ catkin_make
$ cd src
$ catkin_create_pkg ssafy_ad rospy std_msgs
$ cd ~/catkin_ws
$ catkin_make
$ source ~/catkin_ws/devel/setup.bash
$ rospack profile

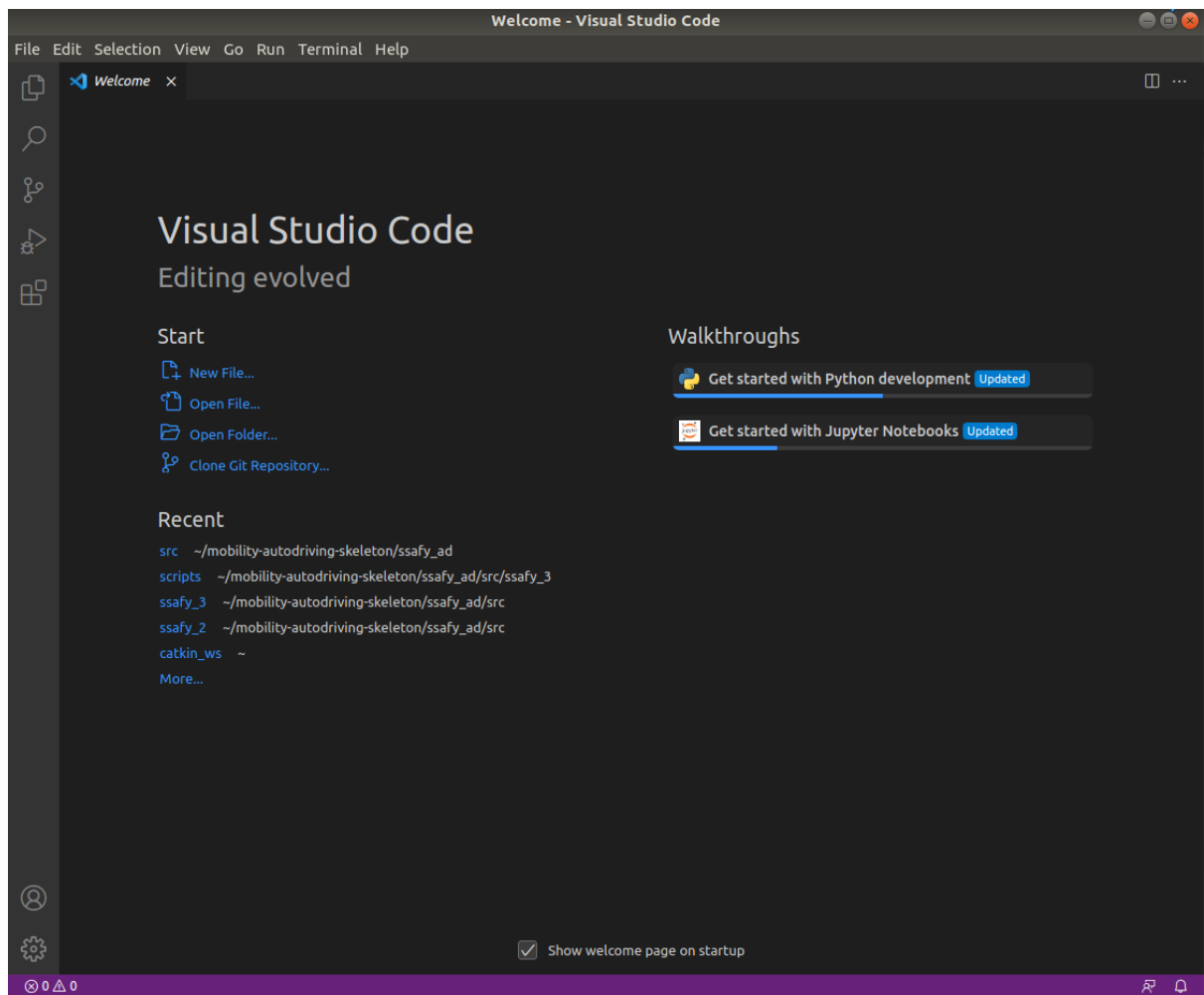
$ sudo apt-get install git
$ cd ~/catkin_ws/src
$ git clone https://github.com/morai-developergroup/morai_msgs.git
$ cd ~/catkin_ws
$ catkin_make
```

1-4. 종속패키지 및 라이브러리 설치


```
$ sudo apt-get install python-pip
$ sudo apt-get install net-tools
$ sudo apt-get install ros-melodic-rosbridge-server
$ sudo apt-get install ros-melodic-velodyne
$ sudo apt install terminator
$ sudo apt install libvulkani
$ pip install pyproj
$ pip install scikit-learn
$ cd ~/catkin_ws
$ catkin_make
```

2-1. Visual Studio Code 설치





2-2. Visual Studio Code Plugin 설치



Python

v2023.6.0

Microsoft [microsoft.com](#) | 81,603,667 | ★★★★★ (535)

IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote), Jupyter Notebooks, code formatting, refactoring, unit tests, and more.

[Disable](#)
[Uninstall](#)
[Switch to Pre-Release Version](#)

This extension is enabled globally.

[DETAILS](#)
[FEATURE CONTRIBUTIONS](#)
[CHANGELOG](#)
[EXTENSION PACK](#)
[RUNTIME STATUS](#)

Python extension for Visual Studio Code

A [Visual Studio Code extension](#) with rich support for the [Python language](#) (for all [actively supported versions](#) of the language: >=3.7), including features such as IntelliSense (Pylance), linting, debugging, code navigation, code formatting, refactoring, variable explorer, test explorer, and more!

Support for [vscode.dev](#)

The Python extension does offer [some support](#) when running on [vscode.dev](#) (which includes [github.dev](#)). This includes partial IntelliSense for open files in the editor.

Installed extensions

The Python extension will automatically install the [Pylance](#) and [Jupyter](#) extensions to give you the best experience when working with Python files and Jupyter notebooks. However, Pylance is an optional dependency, meaning the Python extension will remain fully functional if it fails to be installed. You can also [uninstall](#) it at the expense of some features if you're using a different language server.

Extensions installed through the marketplace are subject to the [Marketplace Terms of Use](#).

Quick start

- **Step 1.** Install a supported version of Python on your system (note: that the system install of Python on macOS is not supported).
- **Step 2.** Install the [Python extension for Visual Studio Code](#).
- **Step 3.** Open or create a Python file and start coding!

Set up your environment

Categories


[Programming Languages](#)
[Debuggers](#)
[Linters](#)
[Formatters](#)
[Other](#)
[Data Science](#)
[Machine Learning](#)
[Notebooks](#)

Extension Resources

[Marketplace](#)
[Repository](#)
[License](#)
[Microsoft](#)

More Info

Published	1/20/2016, 00:03:11
Last released	3/31/2023, 19:25:09
Last updated	3/31/2023, 13:48:20
Identifier	ms-python.python



ROS

v0.9.2 Preview

Microsoft [microsoft.com](#) | 512,369 | ★★★★★ (10)

Develop Robot Operating System (ROS) with Visual Studio Code.

[Disable](#)
[Uninstall](#)

This extension is enabled globally.

[DETAILS](#)
[FEATURE CONTRIBUTIONS](#)
[CHANGELOG](#)
[DEPENDENCIES](#)
[RUNTIME STATUS](#)

Visual Studio Code Extension for ROS

The [Visual Studio Code Extension for ROS](#) provides support for [Robot Operating System \(ROS\)](#) development for ROS1 and ROS2 on Windows and Linux.

Features

- Automatic ROS environment configuration.
- Allows starting, stopping and viewing the ROS core status.
- Automatically create `catkin_make` or `catkin build` build tasks.
- Create catkin packages using `catkin_create_pkg` script or `catkin create pkg`.
- Run `roslaunch` or `roslaunch`.
- Resolve dependencies with `roscdep` shortcut
- Syntax highlighting for `.msg`, `.urdf` and other ROS files.
- Automatically add the ROS C++ include and Python import paths.
- Format C++ using the `ROS clang-format` style.
- Preview URDF and Xacro files.
- Debug a single ROS node (C++ or Python) by [attaching to the process](#).
- Debug ROS nodes (C++ or Python) [launched from a .launch file](#).

Commands

You can access the following commands from the [Visual Studio Code command pallet](#), typically accessed by pressing `ctrl + shift + p` and typing the command name you'd like to use from the table below.

Categories

[Debuggers](#)
[Other](#)

Extension Resources

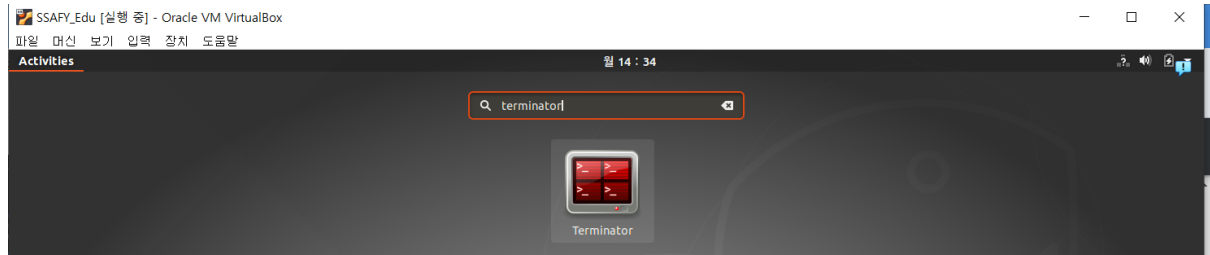
[Marketplace](#)
[Repository](#)
[License](#)
[Microsoft](#)

More Info

Published	6/29/2019, 06:09:22
Last released	4/1/2023, 06:21:58
Last updated	4/3/2023, 10:36:50
Identifier	ms-ros2-vscode-ros

3-1. Terminator 설치

```
$ sudo apt install terminator
```

E. 실행

```
$ cd
$ python main.py

# /SelectCar의 선택하기 버튼(DB : Reservation->choice_btn->choice_btn) 누를 시 실행되는 명령어
$ roslaunch watta_dir watta.launch

# /SelectCar의 예약확인 버튼(DB : Reservation->reservation_btn->reservation_btn) 누를 시 실행되는 명령어
$ rosrn watta_dir autodriving.py

# /CompleteCar의 운행종료 모달의 확인 버튼(DB : Reservation->return_btn->return_btn) 누를 시 실행되는 명령어
$ roslaunch watta_dir watta.launch
$ rosrn watta_dir autodriving.py
```

4 데이터 통신

A. FE to DB

1. 사용자 현재 좌표
2. 목적지 좌표
3. 차량 call 기능 (flag 값)
4. 차량 도착 기능 (flag 값)

B. EMB to DB

1. GPS기반 Ego차량 위치 좌표
2. 목적지 도착 알림 기능 (flag 값)
3. 목적지와의 거리값

C. DB to FE

1. 2-1값을 받아와 web 상에서 차량이 이동하는 경로를 map에 표시
2. 2-2값을 통해 도착 완료 표기

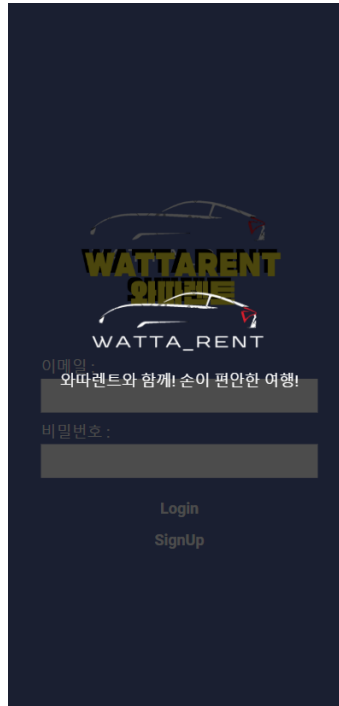
D. DB to EMB

1. 1-3값을 통해 차량 출발
 - case a) 해당 값이 true, 목적지 좌표가 가능할 때
 - case b) 해당 값이 true, 목적지 좌표가 불가능할 때
 - case c) 해당 값이 false일 때

2. 1-1, 1-2값을 바탕으로 최단경로로 이동(Dijkstra)
3. 목적지 부근 도착시 도착 알림 flag 전송
4. 사용자가 운행종료 버튼 누르면 1-4값 확인해서 차고지(start point 고정값)로 다시 이동

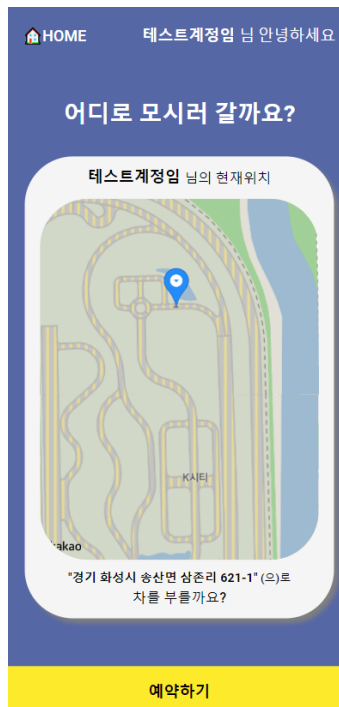
5 시나리오

A. 홈 화면

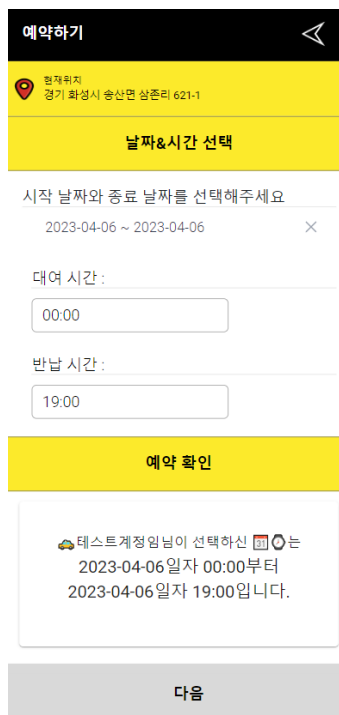


- 시작화면이다.
- 화면을 누르면 로그인 화면으로 넘어간다.

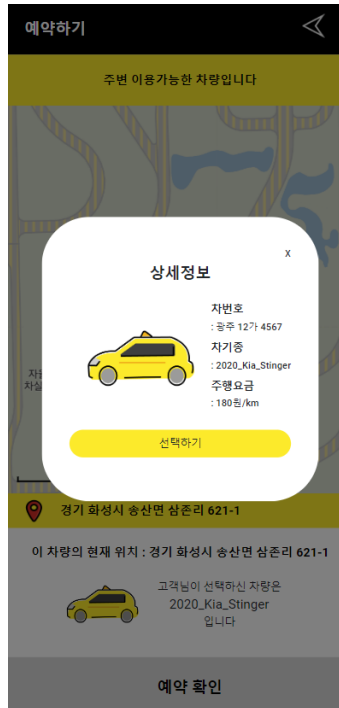
B. 메인 화면



- 차량 픽업받을 장소를 설정하는 페이지이다.



- 예약 날짜 시간을 설정하는 페이지이다.



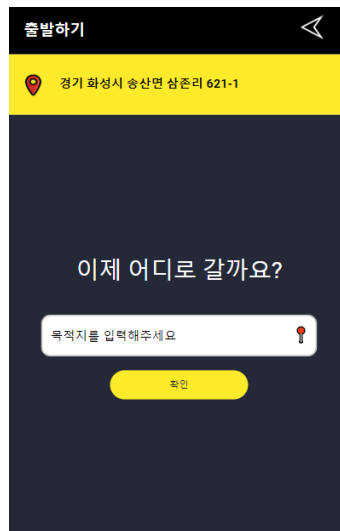
- 차량 선택 화면이다

C. 차량 픽업 화면



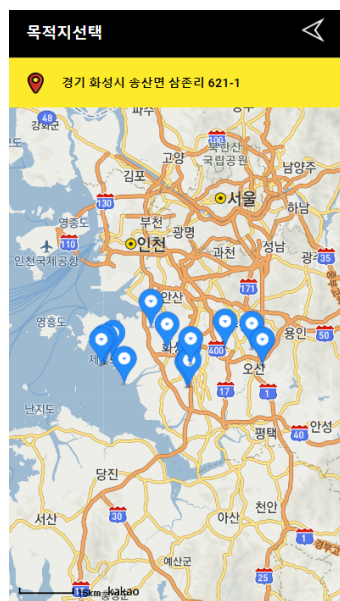
- 차량 픽업 화면이다.

D. 목적지 설정 화면



현재 사용중인 차량
 차량번호:
 차량기종:

운행종료

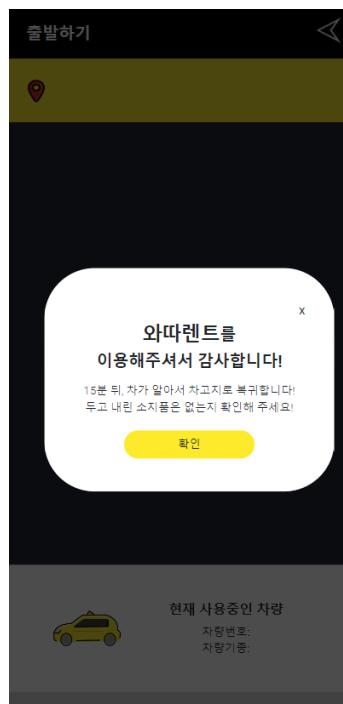


테스트용계정님
 목적지를 선택해주세요!

- 주소를 검색하고 마커를 선택해 목적지를 설정한다.



- 목적지로 이동하는 차량의 모습이다



- 반납 버튼을 누른 모습이다.