

제9기 SW마에스트로과정 예비연수생 프로젝트 기획 최종보고서 요약본

□ 프로젝트 요약

프로젝트 명	공공데이터를 이용한 장거리 운전자를 위한 졸음운전 방지 동승자 봇 어플리케이션
프로젝트 요약	졸음 방지를 위한 동승자 봇 어플리케이션
프로젝트 키워드	교통사고, 공공데이터, 졸음 방지, 봇, 운전자 연결
예비연수생 명	김태정
목적 및 필요성	<ul style="list-style-type: none">- 고속도로의 운전자 중 10명 중 4명은 졸음운전을 경험한 적이 있음- 교통사고 사망 1위의 원인은 졸음운전이며 화물차 고속도로 사망사고의 82%가 졸음과 전방 주시 태만이며 졸음운전 사고의 사망률은 전체 사망률의 2배- 유사 서비스들이 존재하지만 여러 가지 이유로 확률과 안정성이 모자란 면이 있으며 금전적인 추가지불을 하는 등의 문제점이 있음- 졸음운전의 사고 중에서 82.5%의 경우 동승자가 존재하지 않았다. 동승자가 존재할 경우 졸음운전 발생률이 낮은 것을 확인할 수 있음
프로젝트 개요	<ul style="list-style-type: none">- 공공데이터와 R을 통계적 기법을 이용하여 졸음 운전상황과 연관된 데이터를 산출, 그리고 각각의 비율을 알아내고 서로 연동하여서 졸음 위험 지역과 안전 운행 위험 지역을 산출해 낸다. 이 데이터는 사용자의 위치를 GPS를 이용해서 그 위치 데이터 인근 지역의 공공데이터로 산출- 사용자는 운전 중이라서 손을 사용할 수 없기에 음성인식과 TTS를 위하여 네이버 클로버를 사용- 동승자 서비스를 동작 시에 사용자 매칭 모드와 사용자 비매칭 모드를 선택할 수 있으며 사용자 매칭 모드를 선택할 시에 다른 솔루션 사용자와 매칭을 시켜준다.- 사용자 비매칭 모드 시에는 미리 봇에 저장된 동작을 시행하여 사용자가 졸음에 대비할 수 있도록 한다.- 사용자 매칭 모드의 유무와 상관없이 백그라운드에서 사용자 환기 시스템이 주기적으로 작동하여서 사용자가 단조로운 운전을 하지 않게 하기 위해 계속해서 유도
수행 방법 및 일정	<ul style="list-style-type: none">- 공공데이터는 각각의 인터넷 포털 및 웹 크롤링을 통해서 수집, 수집하는 웹 크롤러는 nodejs로 제작- 공공데이터와 받은 GPS위치 데이터는 R을 이용한 통계적 검정으로 가공- 음성인식과 TTS는 네이버 클로버를 이용- 사용자 매칭 시스템은 현재 솔루션을 동작하는 다른 사용자를 랜덤으로 매칭- 사용자 매칭 시스템이 끝난 후 서로 친구를 등록할 수 있으며 나중에 매칭시 랜덤하게 매칭하는게 아니라 친구로 등록한 사람 중에서 매칭 가능- 사용자 비매칭 시스템은 봇 솔루션이 동작- 봇 솔루션에는 여러 행동들이 이미 내장되어있고 선택 가능
기대 효과 및 활용 분야	<ul style="list-style-type: none">- 사용자는 졸음운전의 위험에서 벗어나서 안전 운전이 가능- 사용자 풀을 활용하여 수익 창출을 모색- 네비게이션등에 어플리케이션을 도입하여 물품 판매 ##- 동승자 행동 등의 api를 제공하여서 다른 개발자의 참여 유도

제9기 SW마에스트로과정 예비연수생 프로젝트 기획 최종보고서

□ 프로젝트 목적 및 필요성

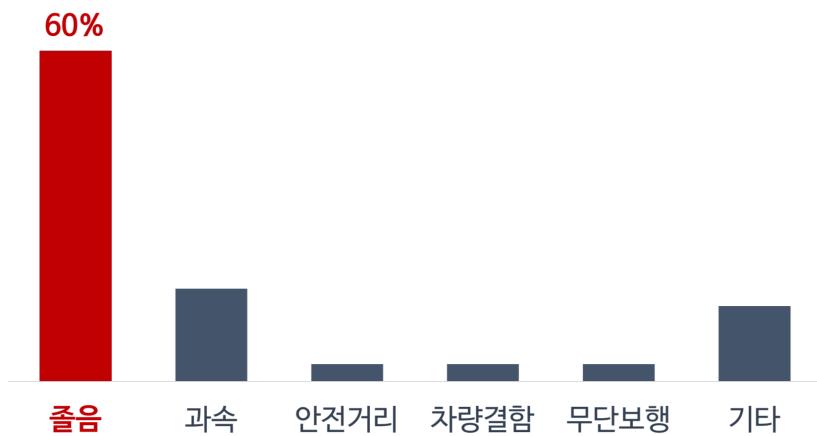
○ 프로젝트의 필요성



[그림 1] 졸음운전에 관한 뉴스

운전을 해본 사람이라면 누구나 운전하면서 피곤하거나 졸린 느낌을 받은 경험이 있을 것이다. 이는 굳이 통계로 나타내거나 할 필요도 없이 한번쯤은 겪어 본 경험이다. 졸음운전은 운전이 자주 반복될수록, 운전 시간이 길어질수록 일어날 확률이 높아진다. 따라서 장거리 운전자는 졸음운전의 위험에 직면해 있다. 일반적으로 사람들은 졸음운전의 위험성을 과소평가하는 경향이 있지만 도로교통공단의 발표에 의하면 졸음운전의 위험률은 음주운전의 2배이다. 졸음운전의 빈도에 대한 구체적인 통계를 언급하자면 고속도로의 운전자 10명 중 4명은 졸음운전을 경험 해본 적이 있다고 답했다. 고속버스 및 화물차 운전자의 95%는 졸음운전을 경험한 적이 있다고 답하였다. 고속도로 교통사고 사망 1위는 졸음운전이며 화물차 고속도로 사망사고의 82%가 졸음과 전방 주시 태만이고 졸음운전 사고의 사망 사고율은 전체 사망 사고율의 2배에 달한다.

2015 상반기 사고건수 교통사고 원인 별 현황 통계



[그림 2] 2015 상반기 사고건수 교통사고 원인 별 현황 통계

대한민국의 장기 운전자의 수를 보자면 운수업자가 1,109,949명이며 2016년 고속도로의 차량의 통행 수는 1,700,286대가 된다. 이렇게 많은 수가 졸음운전에 직접적으로 노출되어 있다. 하지만 아직 마땅한 현실적인 대책이 없는 상황이다. 여러 전문가들은 이에 대한 위험을 지적한다. 졸음운전 방지 기능이 자율주행과 더불어서 앞으로 자동차에 추가 되어야할 새로운 기능 중 하나라고 보고 있다. 이에 여러 기업들이 졸음 방지 솔루션을 차에 도입하는 것을 고려하고 있다.

○ 기존의 졸음 예방 서비스



[그림 3] 마이카 버디 어플리케이션

주기적으로 알람을 울리는 방식으로 대표적으로 마이카 버디가 존재한다. 마이카 버디의 경우 특정 시간 간격이 있으면 그 시간간격으로 지속적으로 알람이 울리기에 이 알람을 끄기 위해서 특정 행동을 취해야 하므로 지속적으로 자극을 줘서 졸지 못하게 하는 방식이다. 그러나 이 방식의 문제점은 정기적으로 알람은 울리지만 졸음 상태 자체를 깨게 하기는 힘들며 시속 100km라고 가정할 경우 3초만 눈을 감아도 80m를 주행하는 고속도로 환경 특성상 알람 사이의 기간 동안 다시 사고의 위험이 있으며 알람의 주기를 짧게 하여 이 문제를 해결하게 하려 할 경우 사용자가 어플리케이션을 사용하면서 상당한 피로를 유발할 수 있다는 단점이 있다.



[그림 4] 뷰메이트

또 다른 방식으로 사람의 얼굴과 눈을 지속적으로 확인하는 아이트래킹을 사용한 방식이며 보통 기기를 사용한다. 이 방법을 채용한 가장 유명한 것이 바로 뷰메이트라는 안면인식, 아이트래킹 카메라이다. 이 기기는 사람의 얼굴을 지속적으로 확인하기에 사람의 졸음을 확인할 수 있는 장점이 있다. 하지만 얼굴을 확인하는 방식이라는 점이 이 뷰메이트의 강점이면서 약점이다. 얼굴을 확인하는 방식이 밤일 경우, 빛이 세게 비치는 경우, 착용자가 안경을 쓴 경우, 머리가 긴 경우 등, 얼굴을 인식하기 힘들지만 흔히 일어나는 상황에서 인식이 힘들다는 단점이 있다.

그 외에도 센서를 이용한 방식들이 시중에 나와 있지만 일단 추가적인 비용이 들고 그 비용을 별개로 생각한다 하더라도 단순히 센서만을 사용한 방식은 인식률이 좋지 않으며 센서를 여러 개를 부착한 방식들의 경우 인식률에 상당한 개선이 있지만 문제는 사용자들이 그런 센서를 다 부착하고 사용하는지에 문제가 있으며 상당부분은 아직 실험단계이다.

○ 프로젝트의 목적

졸음운전이 교통안전에 미치는 문제점은 크다. 그렇지만 기존의 졸음운전 솔루션들은 졸음 상태를 크게 개선하지 못하는 경우가 많으며 센서들을 이용한 방식은 인식률이 크게 기대하지 못하지만 추가비용이 드는 등의 문제점이 있다.

한국도로공사에서는 졸음운전에 대한 해결책으로 5가지를 제시하였으며 그 중에는 동승자와 대화하기가 있다. 동승자와 대화를 하다 보니 졸지 않고 계속 운전을 하게 된 경험이 있을 것이다. 이것은 단순히 경험이 아니라 통계로 말해준다. 졸음운전 사고에서 동승자가 존재함에도 일어난 경우는 28% 밖에 되지 않았다. 여기서 동승자가 깨어있는지의 유무는 조사에 포함되지 않았으므로 실제로는 더욱 높은 확률일 것이다. 그리고 법원의 판례에 따르면 졸음운전에 동승자의 중요도가 30%가 있다고 보고 있다. 동승자의 존재가 졸음운전 사고의 감소에 직접적인 영향을 주는 것이다.

화물트럭운전사와의 인터뷰에서 화물트럭운전사들이 졸음을 깨기 위해 자주 사용하는 방식이 무전이라고 대답하였으며 버스 운전사들의 경우 졸음에 깨기 위하여 전화를 사용한 경험이 존재한다고 답하였다. 이에 의하여 기존의 방식의 문제점을 해결하기 위하여 졸음 상태를 미리 감지하여 그 상황에 마치 동승자가 존재하는 듯 한 느낌을 주어 졸음운전을 예방하고 안전운전을 하게 도움을 주는 솔루션을 개발하려고 한다.

□ 프로젝트 개요

○ 프로젝트 소개



[그림 5] 공공 데이터 API

본 솔루션에서는 공공 데이터를 사용하여 사고의 위험성을 감지한다. 공공 데이터는 공공 데이터 포털이나 한곡도로공사 등의 open api를 사용한다. 출음 사고에 직접적인 영향을 줄 수 있는 요소는 크게 보면 3가지의 요소가 있으며 각각 공간적 위험구간(SDZ), 시간적 위험구간(TDZ), 인적 위험구간(HDZ)이다. 여기서 공공데이터 open api를 이용하여 공간적 위험구간과 시간적 위험구간을 도출해낸다.



[그림 6] 공간적 위험구간 중 하나인 직선도로

공간적 위험구간은 여러 가지 원인으로 판단할 수 있다. 더 정확하게 알기 위해서는 통계적인 검증이 필요하겠지만 현재는 확실히 결정된 요인 중에 출음 사고를 유발하는 1순위의 공간적 위험 요인은 바로 직선 도로이다. 변화가 많은 도로에서는 출음운전을 하기가 힘들다. 집중 하여야하기 때문이다.

분부명	자치구	사고일자	출발구분	사고시간	사고거리(1km)	노선명	이정	방향	사망	부상	총상	경상	발생지점	본관	사고부위별	주 사고원인	사고직접	총인자	교통	환경	사고시	도로구간	요일	날짜	
수도권북부	구로구	2012/01/01	이동	1:40	1~2	영등포	4:00	교통	0	0	0	0	본관	분선차로	3/3차로	88	총동	0	0	0	0	정상	일요일	00:00	
대구경북부	영주시	2012/01/01	이동	3:00	3~4	동명면	212.52	부산	0	0	0	0	본관	분선차로	2/2차로	102	총동	0	0	0	0	정상	일요일	00:00	
대구경북부	구미시	2012/01/01	이동	5:25	5~6	영동면	151.30	부산	0	0	0	0	본관	분선차로	1/2차로	103	총동	0	0	0	0	정상	일요일	00:00	
수도권북부	구리시	2012/01/01	이동	6:25	6~7	동명면	20.70	충북	0	0	0	0	본관	분선차로	1/2차로	104	총동	0	0	0	0	정상	일요일	00:00	
광주전남부	구례군	2012/01/01	이동	7:05	7~8	영동면	36.20	한국	0	0	0	0	본관	분선차로	1/2차로	105	총동	0	0	0	0	정상	일요일	00:00	
수도권북부	구리시	2012/01/01	이동	7:50	8~9	서울영동고속도로(한전고속도로)	16.20	인계동	1	0	0	0	본관	분선차로	1/4차로	106	총동	0	0	0	0	정상	일요일	00:00	
부산경남부	진주시	2012/01/01	이동	6:10	9~10	남해간(승용차부산)	40.80	순천	0	0	0	0	본관	분선차로	1/2차로	91	총동	0	0	0	0	정상	일요일	00:00	
대전충청부	간천군	2012/01/01	이동	6:55	10~11	충북간(승용대전)	282.05	제천	0	0	0	0	본관	분선차로	0	0	총동	0	0	0	0	정상	일요일	00:00	
충청부	밀양군	2012/01/01	이동	8:10	12~13	충북간(승용대전)	51.00	복산	0	0	0	0	본관	분선차로	2/2차로	124	총동	0	0	0	0	정상	일요일	00:00	
수도권북부	수원시	2012/01/01	이동	8:35	13~14	충북간(승용대전)	2.00	제천	0	0	0	0	본관	분선차로	0	0	총동	0	0	0	0	정상	일요일	00:00	
부산경남부	강성군	2012/01/01	이동	4:25	4~5	충북간(승용대전)	78.40	봉양	0	2	0	0	본관	분선차로	1/2차로	9	총동	0	0	0	0	정상	일요일	00:00	
수도권북부	수원시	2012/01/01	이동	4:50	5~6	서해안선	33.00	노고부	0	0	0	0	본관	분선차로	0	0	총동	0	0	0	0	정상	일요일	00:00	
경기판문부	수원시	2012/01/01	이동	7:32	7~8	충북간(승용대전)	217.00	영동	0	0	0	0	터널	내부	0	0	총동	0	0	0	0	정상	일요일	00:00	
경기도	충주시	2012/01/01	이동	7:55	9~10	충북간(승용대전)	208.15	영동	0	0	0	0	본관	노고부	고개	0	총동	0	0	0	0	정상	일요일	00:00	
수도권북부	수원시	2012/01/01	이동	10:42	11~12	충북간(승용대전)	30.40	당동	0	0	0	0	본관	분선차로	4/4차로	113	총동	0	0	0	0	정상	일요일	00:00	
경기판문부	수원시	2012/01/01	이동	10:46	12~13	충북간(승용대전)	371.30	영동	0	0	0	0	본관	분선차로	4/4차로	102	총동	0	0	0	0	정상	일요일	00:00	
경기판문부	수원시	2012/01/01	이동	10:46	16~17	충북간(승용대전)	371.30	영동	0	0	0	0	본관	분선차로	4/4차로	102	총동	0	0	0	0	정상	일요일	00:00	
경기판문부	수원시	2012/01/01	이동	15:00	15~16	서울영동	14.00	임영	0	0	0	0	본관	분선차로	0	0	총동	0	0	0	0	정상	일요일	00:00	
부산경남부	진주시	2012/01/01	이동	21:15	21~22	울진간	8.10	울산	0	0	0	0	본관	분선차로	1/2차로	105	총동	0	0	0	0	정상	일요일	00:00	
대전충청부	보은군	2012/01/01	이동	0:33	0~1	보은간(승용대전)	74.00	금강	0	1	0	1	본관	분선차로	2/2차로	103	총동	0	0	0	0	정상	일요일	00:00	
대구경북부	구미시	2012/01/01	이동	4:00	4~5	경부간(승용대전)	164.10	서울	0	1	0	1	본관	분선차로	4/4차로	107	총동	0	0	0	0	정상	일요일	00:00	
대구경북부	경주시	2012/01/01	이동	4:45	5~6	경부간(승용대전)	248.40	경주	0	0	0	0	본관	분선차로	0	0	총동	0	0	0	0	정상	일요일	00:00	
대전충청부	세종시	2012/01/01	이동	10:00	10~11	경부간(승용대전)	248.40	경주	0	0	0	0	본관	분선차로	0	0	총동	0	0	0	0	정상	일요일	00:00	
경기판문부	경주시	2012/01/01	이동	1:30	1~2	영동간	308.20	충청	0	0	0	0	본관	분선차로	90	90	총동	0	0	0	0	정상	일요일	00:00	
경기판문부	경주시	2012/01/01	이동	1:40	1~2	충북간(승용대전)	203.00	영동	0	0	0	0	본관	분선차로	1/2차로	122	총동	0	0	0	0	정상	일요일	00:00	
경기판문부	구미시	2012/01/01	이동	2:50	2~3	경부간	178.12	부산	0	0	0	0	본관	분선차로	115	총동	0	0	0	0	정상	일요일	00:00		
부산경남부	부산시	2012/01/01	이동	3:02	3~4	경부간	30.70	영동	0	0	0	0	본관	분선차로	4/4차로	100	총동	0	0	0	0	정상	일요일	00:00	
경기판문부	부산시	2012/01/01	이동	4:50	4~5	서울영동	12.00	제천	0	0	0	0	본관	분선차로	1/2차로	103	총동	0	0	0	0	정상	일요일	00:00	
경기판문부	경주시	2012/01/01	이동	5:10	5~6	경북간(호남간)	86.30	순천	0	0	0	0	본관	분선차로	0	0	총동	0	0	0	0	정상	일요일	00:00	
경기판문부	경주시	2012/01/01	이동	8:20	6~7	충북간(승용대전)	12.70	영동	0	0	0	0	터널	입구	0	0	총동	0	0	0	0	정상	일요일	00:00	
경기판문부	경주시	2012/01/01	이동	21:00	21~22	고령영동	30.22	고령	0	0	0	0	본관	분선차로	2/2차로	0	0	총동	0	0	0	0	정상	일요일	00:00
부산경남부	영동면	2012/01/01	이동	1:20	1~2	남해간(승용부산)	114.40	부산	0	0	0	0	본관	분선차로	1/2차로	97	총동	0	0	0	0	정상	일요일	00:00	
수도권북부	시흥시	2012/01/01	이동	1:55	2~3	제2경인선	131.10	인천	0	0	0	0	본관	분선차로	0	0	총동	0	0	0	0	정상	일요일	00:00	
경기판문부	부평면	2012/01/01	이동	2:20	2~3	경부간	145.40	경주	0	0	0	0	본관	분선차로	1/2차로	93	총동	0	0	0	0	정상	일요일	00:00	
부산경남부	영성면	2012/01/01	이동	3:20	3~4	충북간(승용대전)	12.70	영동	1	0	0	0	본관	분선차로	1/2차로	100	총동	0	0	0	0	정상	일요일	00:00	
대구경북부	구미시	2012/01/01	이동	6:23	6~7	경부간	209.65	부산	0	1	0	1	본관	분선차로	2/3차로	101	총동	0	0	0	0	정상	일요일	00:00	
전북도	보령면	2012/01/01	이동	21:10	21~22	서해안선	197.00	서울	0	0	0	0	본관	분선차로	1/2차로	96	총동	0	0	0	0	정상	일요일	00:00	
전북도	장수군	2012/01/01	이동	22:40	22~23	남해간(승용부산)	22.40	광주	0	0	0	0	TG터미널스	고속도로	2/2차로	100	총동	0	0	0	0	정상	일요일	00:00	
수도권북부	경기남부	2012/01/01	이동	0:25	0~1	경부간	321.80	부산	0	0	0	0	본관	분선차로	3/4차로	100	총동	0	0	0	0	정상	일요일	00:00	
부산경남부	진주시	2012/01/01	이동	2:15	3~4	남해간(승용부산)	71.80	부산	0	0	0	0	본관	분선차로	0	0	총동	0	0	0	0	정상	일요일	00:00	
수도권북부	여현면	2012/01/01	이동	6:05	6~7	영동간	49.00	영동	0	0	0	0	본관	분선차로	0	0	총동	0	0	0	0	정상	일요일	00:00	

[그림 7] 2012 ~ 2016 출음 운전 사고 데이터

또한 다른 공간적 위험 요인으로는 졸음 사고 지역이 있다. 이는 더욱 직접적인 데이터로 졸음운전이 발생한 지역 역시 공간적 위험 요인으로 분류할 수 있다. 그 외에도 사고 다발구간 등의 데이터 들이 있다.

순위	전체		봄		여름		가을		겨울	
	시간단위	사고건수								
1	5~6	421	4~5	107	4~5	114	5~6	104	6~7	109
2	4~5	413	5~6	103	5~6	112	4~5	99	5~6	102
3	14~15	368	14~15	100	14~15	111	15~16	94	4~5	93
4	6~7	367	2~3	97	0~1	104	3~4	93	15~16	82
5	2~3	360	16~17	86	3~4	104	14~15	89	1~2	81
6	15~16	355	1~2	85	1~2	103	2~3	87	2~3	81
7	3~4	349	15~16	83	15~16	96	6~7	82	3~4	72
8	1~2	344	6~7	82	2~3	95	7~8	77	7~8	72
9	0~1	301	3~4	80	6~7	94	1~2	75	8~9	70
10	7~8	293	0~1	70	13~14	87	16~17	69	0~1	68
합계		6190		1593		1705		1460		1432

[그림 8] 시간별, 계절별 사고 발생 현황

시간적 위험 구간으로는 사고 발생 시간대와 사고 발생 계절 대를 확인하는 방법이 있다. 현재 도출된 자료로 보면 사고 발생 시간대로 보았을 때는 심야 시간대에 확실히 사고가 자주 일어나며 봄, 여름 등의 나른한 날씨 대에 사고가 확실히 많이 일어나는 것으로 산출되었다.

이렇게 open api를 통하여 공간적, 시간적 위험 요소를 도출할 수 있다. 여기에 소개 되지 않은 수많은 요인들이 있다. 이러한 데이터를 단순히 통계 그래프를 확인하거나 임의로 취사선택하는 것이 아니라 open api와 통계 프로그래밍(R이나 python등을 사용)을 활용하여 위험요소를 통계적으로 검증하고 이를 바탕으로 졸음운전 위험지역을 산출해낸다. 중요한 것은 사용자가 해당지역에 진입하기 전에 사용자를 위치를 기반으로 하여 미리 데이터를 수집하여 위험지역을 알아내는 것이 중요하다.

○ 어플리케이션의 동작

모바일에서 어플리케이션이 항상 백그라운드에서 동작한다. 어플리케이션이 동작 중이면서 해당 위험 지역에 도달하였을 경우 서비스가 자동으로 동작한다. 이때 봇은 해당 지역이 위험 지역임을 알린다. 서비스는 봇 서비스로서 동작한다. 봇은 보이스 봇으로 음성으로 사용자에게 정보를 제공한다. 운전을 하면서 손을 사용하는 것은 위험하기 때문이다.



[그림 9] 봇 서비스의 졸음 유발 지역 알림

해당 위험 지역에 진입하였음을 알린 후 봇은 사람과 매칭을 할지 아니면 계속해서 봇모드로서 동작할지를 물어본다. 매칭을 하는 대상은 자신이 아는 사람으로 자신의 모바일의 전화번호부에 등록되어 있거나 해당 어플리케이션을 사용하는 또 다른 그룹, 혹은 이 어플을 사용하는 다른 사용자와 랜덤하게 매칭할 수 있으며 사용자는 이를 선택할 수 있다.



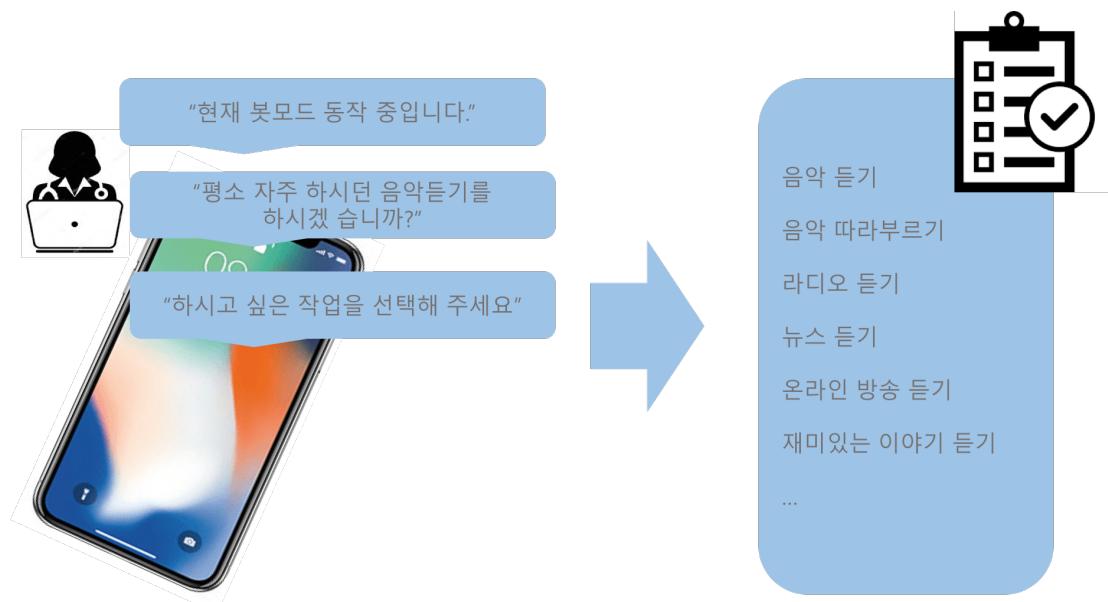
[그림 10] 봇모드와 사용자와의 채팅모드 선택

옆에서 동승자가 말동무가 되어 주면서 운전하는 것이 출음운전을 예방하는데 큰 도움이 되므로 이 어플리케이션을 사용하는 다른 장거리 운전자들끼리도 랜덤하게 매칭시킬 수 있는 기능을 넣은 것이다. 실제로 사고가 일어나는 시간대에서는 같은 장거리 운전자가 아니라면 연락할 수 없는 경우가 많기 때문이다. 물론 이는 옵션이다.



[그림 11] 랜덤 매칭 종료 이후

만약 랜덤 매칭을 사용하였다면 해당 사용자를 저장하고 등록할 수도 있다. 이는 물론 상대편에서 수락을 했을 경우에만 가능하다. 서로 운전하는 시간대가 비슷하고 이야기가 잘 통한다면 사용자를 등록하여서 서로 말동무가 되어줄 수 있게 하는 것이다. 이 어플리케이션은 이러한 운전자들끼리 커뮤니케이션을 형성하는 것을 권장한다.



[그림 12] 봇모드의 동작과 기능 추가

사람과의 대화를 원하지 않거나 매칭을 할 사람이 딱히 없을 경우 자동으로 봇모드로 넘어간다. 봇이 사람을 대신해서 마치 동승자처럼 주위를 환기시켜주는 것이다. 그렇다고 계속해서 봇과 사람이 대화를 하는 것이 아니라 봇이 사용자가 졸리지 않도록 다른 작업들을 제시해 준다. 여기서는 이를 졸음 방지 봇 액션 리스트라고 부른다. 이는 여러 종류가 있으며 추후에 계속해서 리스트를 추가할 수 있다.

봇 모드에서 선택할 수 있는 작업(졸음 방지 봇 액션 리스트)들은 사용자에게 주위를 환기 시켜서 사용자가 졸리지 않게 하는 것이다. 이 작업은 자주 선택하거나 마지막으로 선택한 작업들을 우선적으로 물어본다. 그 작업을 선택하지 않는다면 봇 모드에서 선택할 수 있는 작업들을 선택할 수 있다. 사용자는 운전 중이기에 손을 사용할 수 없다. 따라서 이 작업은 마찬가지로 음성인식으로 작업을 선택할 수 있다.



[그림 13] 주위를 계속 환기시키기 위한 솔루션

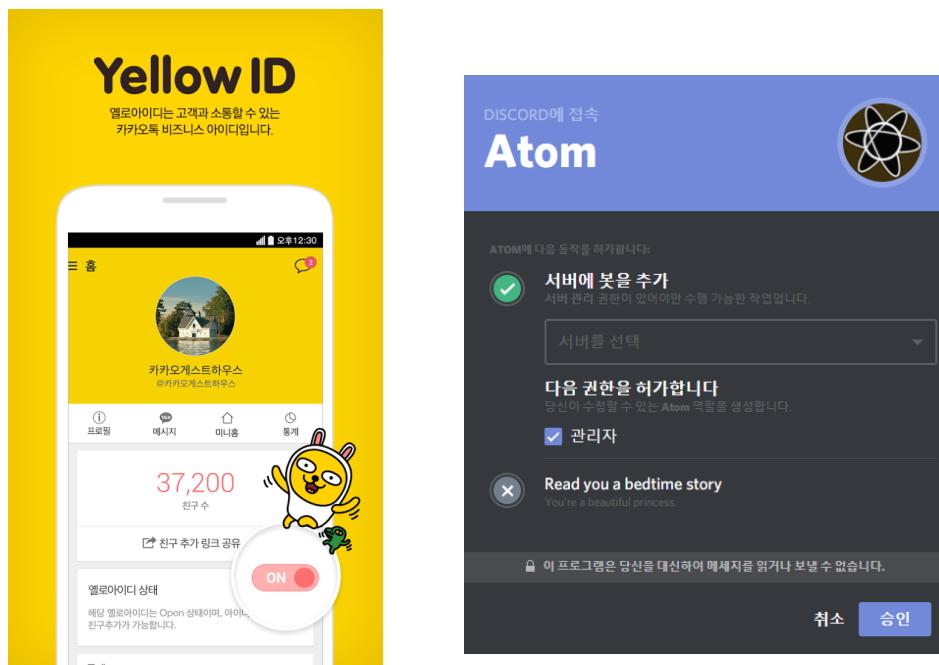
이 솔루션에서 가장 중요하게 생각하는 점은 졸음운전을 하지 않게 하기 위하여 사고를 방지하는 것이다. 그렇기에 운전자에게 주위를 계속해서 환기시켜주는 것 또한 매우 중요하다. 졸음운전의 직접적인 원인들의 공통점이 적은 변화라는 점에서 주위환경은 중요하다. 따라서 이 솔루션에서는 계속해서 백그라운드에서 어플리케이션이 동작하면서 사용자의 주위를 환기시켜준다. 가령 대화가 매칭이 되었을 경우에는 사용자가 대화에 과몰입 하지 않게 하기 위해서 알람을 준다. 또한 운전 중에 공기를 순환시켜서 졸음을 깨는데 도움을 주는 등을 역할을 한다. 지속적으로 사용자를 관리하여서 안전운전을 하게 하는 것이다.

○ 졸음 방지 봇 액션 리스트의 추가



[그림 14] 졸음 방지 봇 액션리스트 추가

졸음 방지 봇 액션리스트는 추후에 개발이 쉽도록 api가 제공된다. 해당 솔루션은 대화형으로 이루어져있고 사진이나 동영상을 보여줄 수 없고 그럴 필요도 없다. 따라서 api를 제공해서 문자열과 사운드 리소스와 코드들을 받아서 액션리스트들을 추가할 수 있다. 일종의 마이크로 어플리케이션을 개발하고 이를 해당 어플리케이션에 추가하는 방식이다. 이는 추후에 개발자들이 리스트에 새로운 액션을 추가하기 수월하게 만들어 준다.

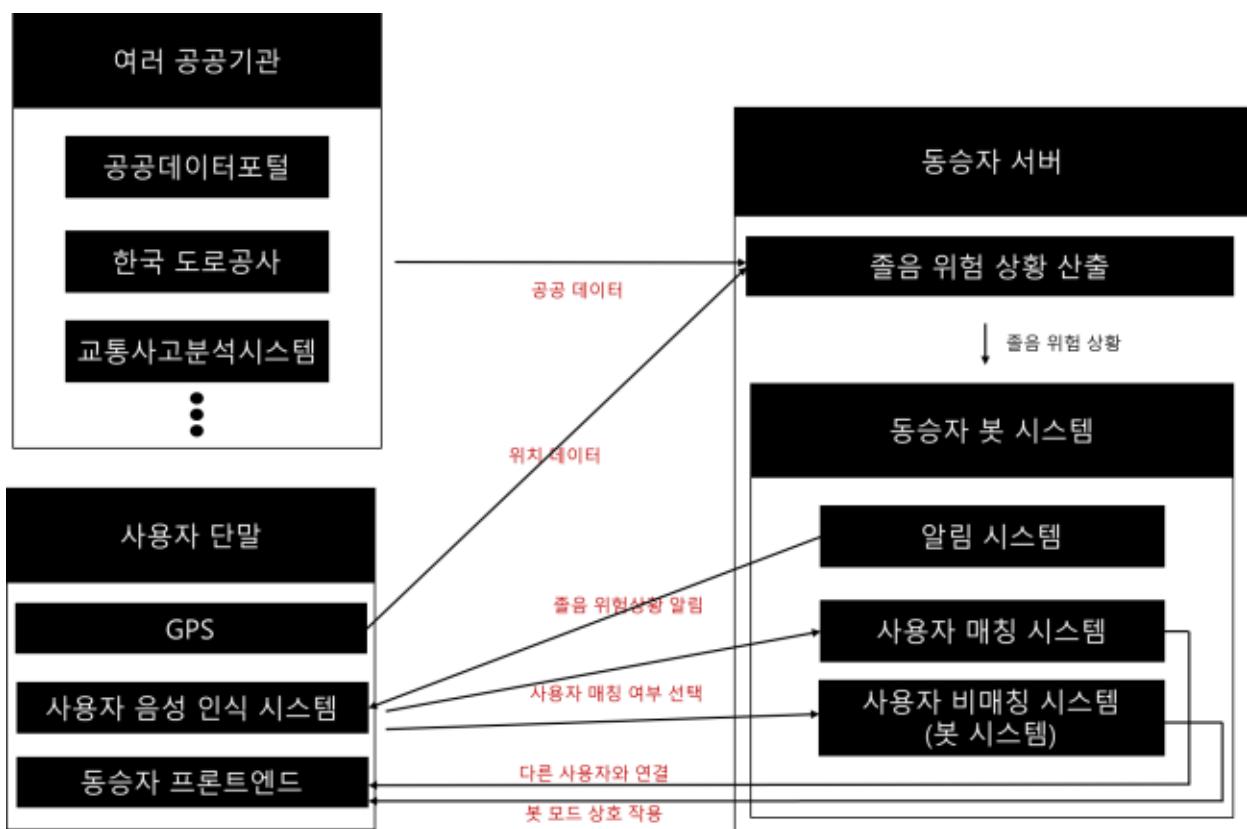


[그림 15] 카카오 봇

[그림 16] 디스코드 봇

기본적으로는 해당 어플을 개발하는 개발자들이 기능을 추가하겠지만 해당 api가 개방되고 나서부터는 기존의 개발자들뿐만 아니라 다른 서드파티 개발자들도 마이크로 어플리케이션을 개발할 수 있다. 해당 리스트는 일종의 앱스토어처럼 인스톨 받을 수 있다. 유료 액션을 추가할 수 있고 사운드 광고를 추가할 수 있어서 해당 방식으로 수익을 낼 수도 있다. 비슷한 사례로는 디스코드 봇이나 카카오 봇같이 api를 통해서 어뎁팅하여 돌아가는 마이크로 어플리케이션들이 존재한다.

○ 시스템 구성도

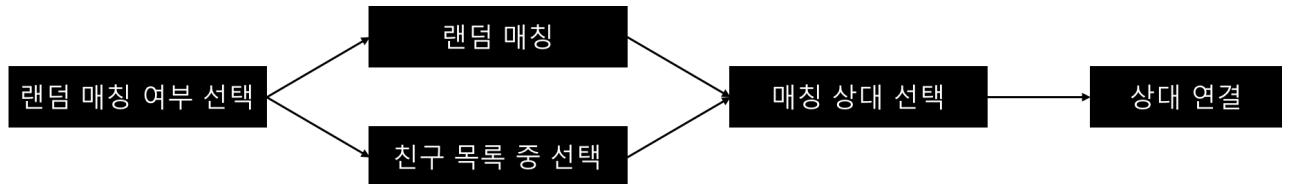


[그림 17] 전체 시스템 구성도

전체 시스템은 위와 같이 구성된다. 공공기관에서 공공데이터를 동승자 서버에 전달한다. 사용자의 단말(모바일 등의 GPS기기를 의미한다.)에서는 위치 정보를 동승자 서버에 전달한다. 이 전달된 데이터를 바탕으로 졸음 위험상황을 산출하게 된다. 졸음 위험 상황은 동승자 서버 내에서 내부의 프로그램으로 산출하게 되며 이 데이터는 졸음 위험상황 데이터로 만들어 지게 된다.

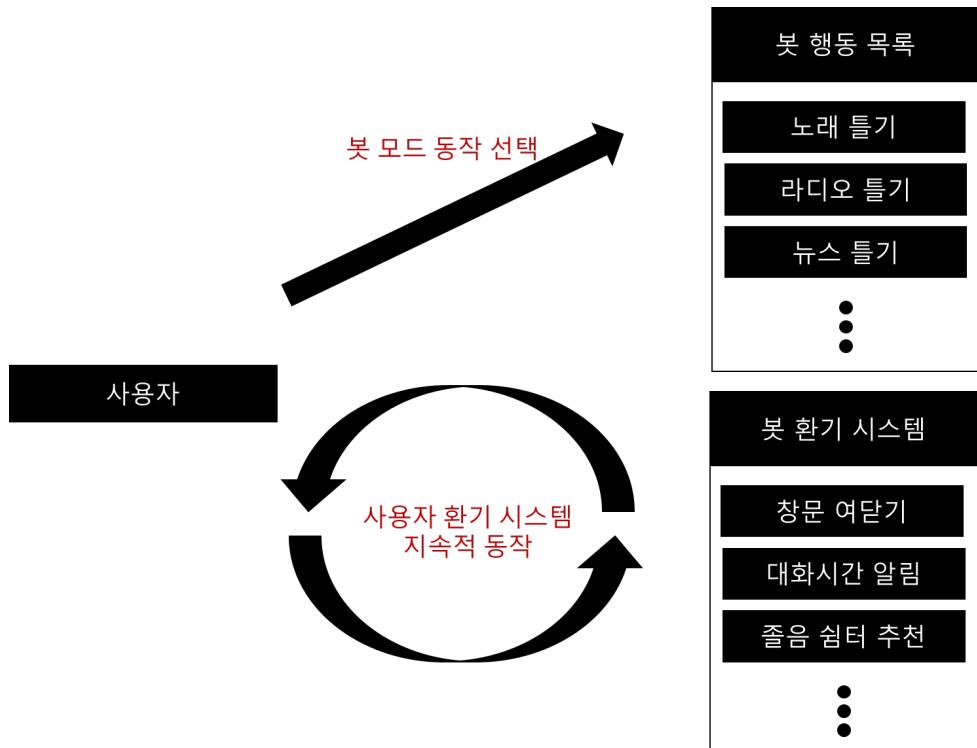
위 과정에서 구해진 데이터를 상시에 받고 있다가 사용자가 졸음운전 위험상황에 진입하기 직전이라는 것이 판단되면 위의 동승자 봇 시스템을 동작 시킨

다. 작동 모드는 총 2가지로 사용자 매칭 시스템과 사용자 비매칭 시스템으로 나누어지며 사용자는 이 두가지중 하나를 선택하여서 작동을 시킬 수 있다. 사용자 매칭을 선택하면 사용자와 연결이 되며 선택하지 않았을 때는 자동으로 봇 모드를 선택하게 된다.



[그림 18] 사용자 매칭 시스템 순서도

사용자 매칭 시스템은 위와 같이 진행된다. 먼저 랜덤 매칭 여부를 선택한다. 그 다음 랜덤 매칭을 선택하였다면 현재 이 어플리케이션을 사용하고 있는 다른 사람과 랜덤하게 매칭을 시켜준다. 만약 이 랜덤매칭을 원치 않는다면 자신의 친구등록에 등록된 사용자나 전화번호부에 등록된 사람들을 대상으로 매칭을 할 수 있고 이는 사용자가 선택할 수 있다. 매칭 상대가 결정되게 되면 상대와 연결이 되어서 서로 대화를 나눌 수 있다.

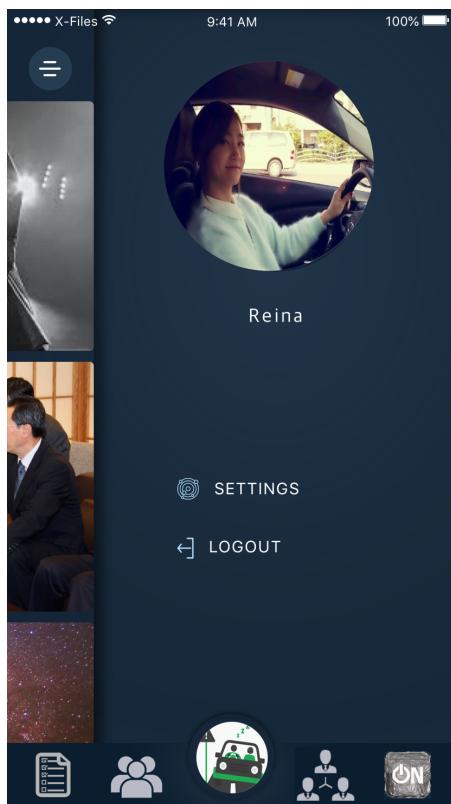


[그림 19] 봇 시스템 동작 로직

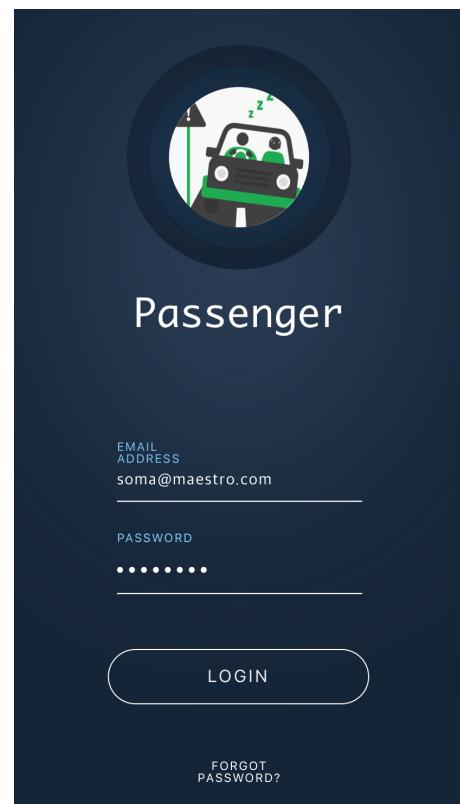
사용자 비매칭 시스템은 봇 시스템을 그대로 사용하는 것을 의미한다. 봇 모드 동작을 선택할 경우 졸음 방지 봇 액션 리스트(이하 봇 행동 목록)에서 미리 정의된 행동들을 선택할 수 있다. 이는 마찬가지로 사용자가 손을 사용해서 선택하는 것은 위험하므로 음성인식을 통해서 선택할 수 있다. 노래를 듣거나 라디오를 듣거나 뉴스를 듣는 등의 액션을 선택할 수 있으며 사용자가 자주 사용하는 봇 목록을 저장하게 되어서 먼저 사용자에게 자주 사용했던 목록을 묻고 그 행동을 우선적으로 실행할 수 있게 한다. 만약 그 행동을 사용하지 않는다면 다시 목록에서 선택할 수 있다. 물론 api를 통해서 제공된 기능을 써서 본인이 행동을 직접적으로 추가하는 것이 가능하며 남들이 만든 행동을 인스톨 받아서 사용하는 것 역시 가능하다.

이 기능들이 돌아가는 와중에도 봇 환기 시스템이 지속적으로 백그라운드에서 돌아가게 된다. 환기 시스템의 목적은 사용자가 자친 지루해 지거나 정적인 상황이 지속되어서 해당 상황에 익숙해져서 졸음이 오는 상황을 방지하게 한다. 기본적으로 한국도로교통공사에서 추천하는 방법인 주기적으로 창문을 여는 것과, 대화시간을 알려주기, 운전에 관한 팁 등을 알려주면서 주위를 환기시키게 한다.

○ 어플리케이션 UI/UX

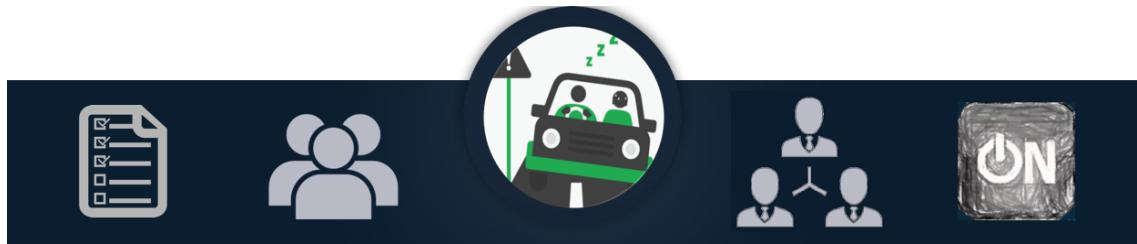


[그림 20] 개인 정보 보기



[그림 21] 어플리케이션 로그인

어플리케이션은 회원가입 후 로그인하여 동작한다. 자신의 개인정보를 볼 수 있다. 어플리케이션은 대부분 음성으로 작동하므로 실제로 어플리케이션을 조작하는 경우는 많지 않다.

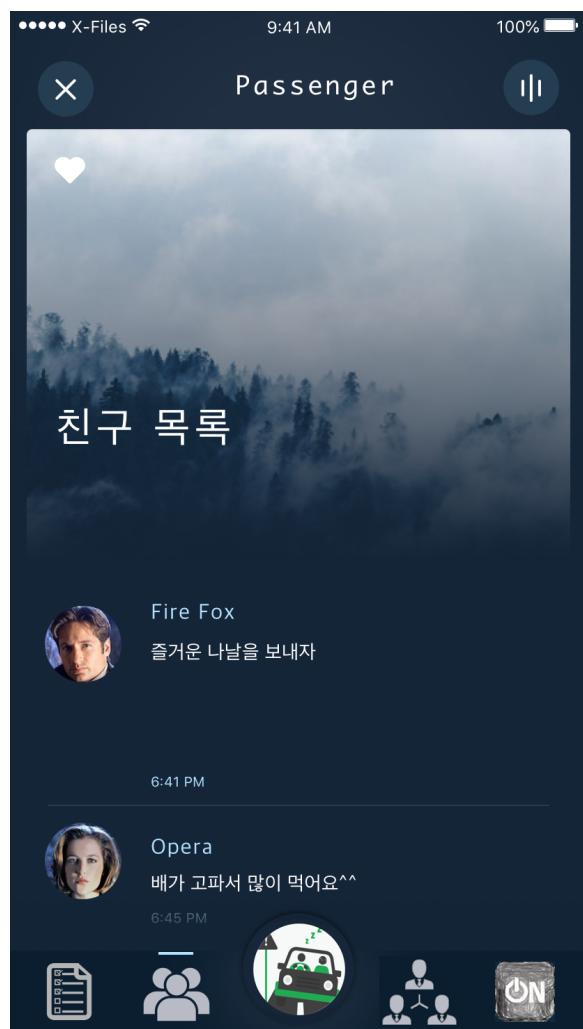


[그림 22] 어플리케이션 네비게이션 바

어플리케이션 네비게이션바에는 총 4가지의 기능이 있다. 제일 처음부터 봇 행동 목록, 친구 목록, 단체 확인, 봇 행동 ON/OFF이다. 봇 행동 모드는 해당 그림의 경우 On 상태이며 한 번 더 클릭해서 Off 상태로 전환 할 수 있다.

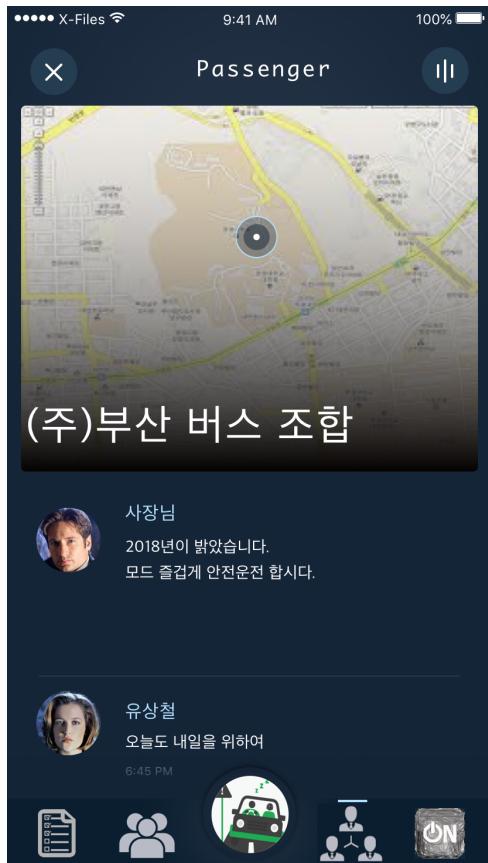


[그림 23] 봇 행동 목록



[그림 24] 친구 목록

봇 행동 목록의 경우 현재 사용자가 사용할 수 있는 행동 목록을 보여준다. 추가적으로 인스톨하거나 본인이 등록할 경우 아래에 추가되며 순서를 변경할 수 있다. 또한 친구 목록의 경우 본인이 전화 기록부에 등록한 사람에 추가적으로 어플을 사용해서 만난 사람이 등록된다.

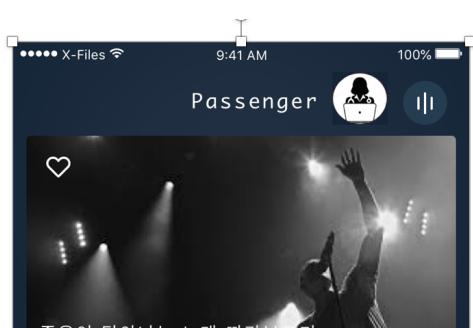


[그림 25] 그룹보기

그룹 보기에서는 현재 본인이 속한 그룹을 확인할 수 있다. 그룹 역시 채팅의 대상이 될 수 있다.



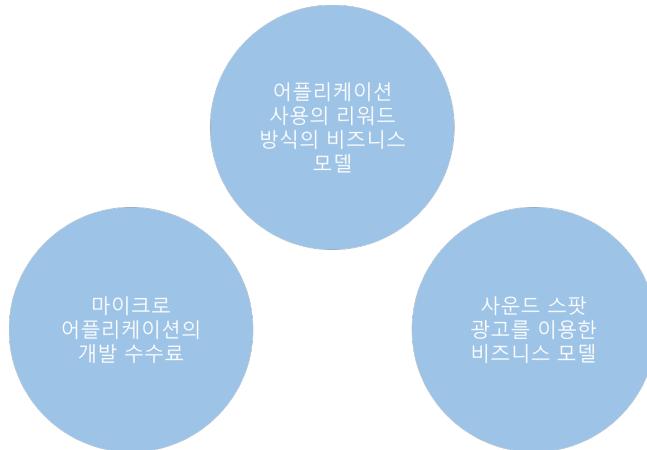
[그림 26] 봇 모드 마크



[그림 27] 봇 모드 동작 화면

실제로 어플을 조작하는 경우는 많지 않다. 보통은 음성인식모드로 동작하고 음성 인식모드로 동작 시에는 위의 봇모드 마크가 떠서 음성모드 동작 중임을 알린다. 위의 모드에서는 음성인식으로 페이지가 이동한다.

○ 비즈니스 모델



[그림 28] 비즈니스 모델

현재 본 솔루션에서는 총 3가지의 비즈니스 모델이 존재한다. 리워드 방식의 비즈니스 모델과 마이크로 어플리케이션의 수수료를 이용한 비즈니스모델, 또한 사운드 스팟 광고를 이용한 비즈니스 모델이 존재한다.



[그림 29]스팟 광고



[그림 30]리워드 모델

리워드는 사용자가 어플을 사용하는 대가로 일종의 보상을 주는 것이다. 리워드광고는 해당 회사와 협약을 맺어서 사용하는 것으로 다른 회사와 거래를 맺어서 리워드를 사용하는 대가로 수익을 얻을 수 있다.

본 솔루션에서의 스팟 광고는 다른 스팟 광고와는 조금 다르다. 대부분의 경우 사운드를 통해서 동작하므로 눈으로 보여주는 이미지 스팟 광고와는 달리 사

운드로 스팟 광고를 사용할 수 있다. 라디오 같은데서 광고를 해주는 개념이라고 생각하면 된다. 사운드 광고를 게재함으로써 해당 업체에 광고비를 받는 수익모델을 수립할 수 있다.



[그림 31] 어플리케이션 스토어

마지막으로는 본 솔루션 위에서 언급 하였던 마이크로 어플리케이션을 개발하여 개제하는 것을 대가로 수수료를 받는 방식이 있다. 이 방식은 기존에 구글이나 애플 등의 회사에서 사용한 앱 스토어와 비슷한 개념으로서 개발자에게는 쉬운 API와 보수를 통해서 추가 개발을 하도록 유도할 수 있고 그로 인해 수수료를 얻음으로써 수익을 유도할 수 있다.

□ 프로젝트 수행 방법 및 일정

○ 주요 기능별 수행방법

사용자 비매칭 시스템 (봇 시스템)	- 랜덤 매칭을 사용하지 않을 시 봇 시스템을 사용 - 봇 시스템은 봇 행동 목록이라는 목록을 가짐 - 봇 행동 목록은 사용자가 추가하지 못하고 추후에 개발자가 행동 목록을 추가할 수 있음 - 봇 모드의 행동은 평소에 자주 사용한 행동과 마지막에 사용한 행동을 기억했다가 다시 시행하시 우선적으로 선택할 수 있도록 함
봇 사용자 환기 시스템 (봇 시스템)	- 봇은 백그라운드에서 계속하여서 사용자가 운전 중인 동안에 사용자의 주의 주위를 환기시키기 위해서 작동함 - 환기 목록은 봇 행동에 따라서 달라질 수 있음 - 봇 환기 목록은 사용자가 추가하지 못함, 추후 개발자 추가 가능
단말(모바일)	- 안드로이드와 ios에서 모두 개발하도록 함 - 안드로이드의 경우 kotlin으로 개발 - ios의 경우 swift를 통하여 개발 - 단말에 음성 인식 api인 클로버를 사용 - 단말은 봇 서버와 통신하여서 지속적으로 데이터를 주고받음 - 단말과 봇 서버가 주고받는 데이터는 restful 디자인으로 주고받음
동승자 서버	- amazon ec2 서버를 사용 - 서버의 언어는 nodejs와 python중 추후에 더 적합한 언어로 사용 - 사용자의 중요한 데이터는 RDBMS이면서 오픈소스인 mariaDB를 사용 - 실시간으로 처리해야 할 데이터들은 NoSQL인 HBase와 인메모리DB인 redis를 사용하도록 함 - 동승자 서버에 사용자 매칭 시스템과 봇 시스템이 모두 들어 있음 - 동승자 서버와 단말과의 데이터 전송은 restful로 주고받음

UI	<ul style="list-style-type: none"> - 어플리케이션은 출음 방지라는 특수한 상황 상 손을 거의 쓰지 않음 - 어플리케이션은 친구목록을 확인하거나 설정을 바꾸는 등의 간단한 기능만 사용할 수 있음 - 어플리케이션에서 동승자 모드에서 자동으로 봇모드와 랜덤 매칭 모드를 선택할 수 있게 하는 등의 선택을 할 수 있음 - UI의 개발은 sketch를 사용하도록 함
공공 데이터 수집	<ul style="list-style-type: none"> - 사용자의 위치를 단말의 GPS모듈로 받는다 - 그 위치를 바탕으로 공공데이터를 수집한다. - 공공데이터는 API를 제공할 경우 그 API를 json으로 받거나 json으로 가공해서 사용한다. - 공공데이터가 특정 API를 제공하지 않을 경우 nodejs의 웹 크롤링을 사용해서 데이터를 정제해서 사용한다.
공공 데이터 가공	<ul style="list-style-type: none"> - 받은 데이터를 바탕으로 R을 이용해서 통계적 검정을 사용하여 필요한 공공 데이터를 산출 - 검정을 통하여 걸러진 데이터를 바탕으로 R을 이용한 통계적 검정을 통하여 받아낸 데이터들이 출음운전에 미치는 비율을 산출 - 검정을 통해서 받아낸 데이터와 그 비율을 바탕으로 출음 위험지역을 산출 - 비 위험지역과 위험지역의 차이를 R을 이용한 통계적 검정을 이용하여 산출
음성 인식 및 TTS	<ul style="list-style-type: none"> - 네이버 클로버를 이용하여서 TTS를 사용하여 음성 서비스를 사용함 - 동승자 서버에서 받은 데이터를 TTS를 이용하여서 사용자에게 상황 및 서비스를 알림 - 네이버 클로버를 이용하여서 음성인식 서비스를 사용함 - 음성인식에 대한 오차를 줄이기 위해서 머신러닝을 통하여 학습하게 함 - 오차 학습에 대한 데이터는 nodejs의 웹 크롤링을 통하여 받음
사용자 매칭 시스템	<ul style="list-style-type: none"> - 랜덤 매칭을 사용할 시 현재 자동차를 타고 있고 랜덤 매칭 시스템을 사용하고 있는 사람과 매칭을 시켜준다. - 자동차를 타고 있는지는 현재의 GPS의 트래킹을 바탕으로 속도를 확인하여 차에 타고 있는 사람을 바탕으로 매칭시켜준다. - 랜덤 매칭을 끝낸 후 사용자를 친구등록 할 수 있다 - 친구 매칭을 선택할 경우 저장되어있는 친구 목록 중에 한명을 선택해서 볼 수 있다.

○ 추진 일정

구분	추진내용	추진일정					
		6월	7월	8월	9월	10월	11월
계획	개발 언어와 공공데이터를 선택						
분석	공공데이터를 분석해서 사용할 자료 취사선택						
설계	개발에 필요할 각각의 솔루션을 설계						
개발	공공데이터를 이용한 솔루션 개발						
	동승자 서버 개발 및 공공데이터 솔루션 연동						
	모바일 개발 및 서버와 연동						
테스트	개발을 진행하는 모든 부분에서 테스트						
종료	마지막으로 테스트 및 UI 마무리						

○ 적용기술

구분		항목	세부 내용
S/W 개발환경	OS	MacOS Android Linux(Ubuntu)	기본적인 개발은 MacOS에서 개발하되 테스팅을 위하여 Android 환경을 사용할 수 있고 서버 테스팅을 위하여 리눅스를 사용할 수 있음
	개발도구	AndroidStudio WebStorm Xcode PyCharm Sketch	안드로이드 개발을 위해서 안드로이드 스튜디오를 사용, 서버 개발을 위하여 WebStorm을 사용하고 ios개발을 위하여 XCode를 사용, 서버를 python으로 선택할 경우 PyCharm을 사용, 모바일 UI개발을 위하여 Sketch를 사용
	개발언어	Kotlin Swift Python JavaScript(nodejs)	안드로이드 개발을 위해서 Kotlin사용, ios개발을 위해서 Swift를 사용, 서버를 개발을 위해서 Python혹은 nodejs를 사용
모바일		android mobile ios mobile	테스트 환경을 위하여 두 종류의 os가 탑재된 모바일을 사용
서버		Amazon aws ec2	사용하는 서버는 아마존은 ec2를 사용
데이터베이스	RDBMS	MariaDB	오픈소스이면서 RDBSMS인 MariaDB를 사용
	NoSQL	Hbase Redis	데이터를 처리하기 위해서 HBase를 사용하고 빠른 연산을 위해서 redis를 사용

□ 기대효과 및 활용분야

사용자 측면	<ul style="list-style-type: none"> - 현대 자율주행 다음으로 이슈인 출음운전은 많은 교통사고를 유발하는 직접적인 원인이고 사용자는 항상 출음운전의 위험에 처해있다. 본인의 의지로 출음 상태에서는 벗어나기 힘들지만 출음 방지 솔루션인 동승자 어플리케이션을 이용하여서 출음 상태에 돌입하지 않고 안전한 운전을 하도록 도움을 줌 - 기존의 출음운전 해결책들은 명확하지 않거나 가격이 많이 드는데 이 솔루션을 사용함으로써 가격 부담을 줄이면서도 기존의 해결책 보다 좋은 효과를 누릴 수 있음
비즈니스 측면	<ul style="list-style-type: none"> - 리워드 모델을 사용하여서 다른 업체와 제휴를 맺어서 이득을 취할 수 있고 사용자에게는 보상을 주어서 어플리케이션 사용을 유도할 수 있음. - 사운드 스팟 광고를 사용하여 다른 업체와 제휴를 맺을 수 있음. - 마이크로 어플리케이션을 다른 개발자 및 개개인이 쉽게 개발하게 만들어서 개발을 유도하거나 그 수수료를 이용하여서 이득을 챙길 수 있음.
개발자 측면	<ul style="list-style-type: none"> - 동승자 행동들을 다양하게 추가함으로써 확정성을 확보할 수 있음, 추후에 새로운 솔루션 추가 시에 손쉽게 추가가 가능하고 api를 제공하여서 다른 개발자의 참여를 유도할 수 있음 - 소셜 네트워크가 결합된 형태를 이용하여서 소셜 네트워크와 관련된 상품을 추가적으로 개발할 수 있음