

제9기 SW마에스트로과정 예비연수생 프로젝트 기획 중간보고서 요약

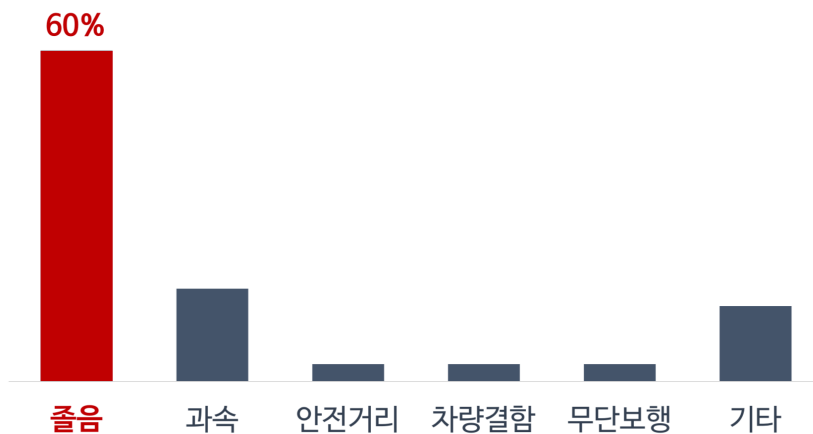
□ 프로젝트 요약

프로젝트명	공공데이터를 이용한 장거리 운전자를 위한 졸음 운전 방지 동승자 어플리케이션
프로젝트 요약	졸음 방지를 위한 동승자 어플리케이션
프로젝트 키워드	교통 사고, 공공데이터, 졸음 방지, 봇, 운전자 연결
예비연수생 명	김태정
목적 및 필요성	<ul style="list-style-type: none"> - 고속 도로의 운전자 중 10명 중 4명은 졸음 운전을 경험한 적이 있음 - 교통사고 사망 1위의 원인은 졸음 운전이며 화물차 고속도로 사망사고의 82%가 졸음과 전방 주시 태만이며 졸음 운전 사고의 사망률은 전체 사망률의 2배 - 유사 서비스들이 존재하지만 여러 가지 이유로 확률과 안정성이 모자란 면이 있으며 금전적인 추가지불을 하는 등의 문제점이 있음 - 졸음 운전의 사고 중에서 62%의 경우 동승자가 존재하지 않았다. 동승자가 존재할 경우 졸음 운전 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 동작시에 사용자 매칭 모드와 사용자 비매칭 모드를 선택할 수 있으며 사용자 매칭 모드를 선택할 시에 다른 솔루션 사용자와 매칭을 시켜준다. - 사용자 비매칭 모드시에는 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 웹크롤링을 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 졸음 운전의 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 중간보고서

□ 프로젝트 목적 및 필요성

2015 상반기 사고건수 교통사고 원인 별 현황 통계



[그림 1] 2015 상반기 사고건수 교통사고 원인 별 현황 통계

○ 프로젝트의 필요성

장거리 운전자는 졸음 운전의 위험에 직면해 있다. 고속 도로의 운전자 10명 중 4명은 졸음 운전을 경험 해본 적이 있다고 답했다. 고속버스 및 화물차 운전자의 95%는 졸음 운전을 경험한 적이 있다고 답하였다. 고속도로 교통사고 사망 1위는 졸음 운전이며 화물차 고속도로 사망사고의 82%가 졸음과 전방 주시 태만이고 졸음 운전 사고의 사망 사고율은 전체 사망 사고율의 2배에 달한다.

대한민국의 장기 운전자의 수를 보자면 운수 업자가 1,109,949명이며 2016년 고속도로의 차량의 통행 수는 1,700,286대가 된다. 이렇게 많은 수가 졸음 운전에 대해 노출되어 있지만 마땅한 대책이 없는 상황이다.

여러 전문가들은 이에 대한 위험을 지적하며 여러 기업들이 졸음 방지 솔루션을 차에 도입하는 것을 고려하고 있다.

○ 유사 서비스

졸음 예방 서비스는 여러 가지 서비스가 현재 존재한다. 크게 나눈다면 세가지의 방식이 있으며 첫 번째로는 주기적으로 알람 혹은 진동을 주어서 사용자에게 환기를 시키는 방식이 있고 두 번째로는 안면 인식 및 아이트래킹을 사용하여 사용자의 졸음을 인지 하게하는 방법이 있고 세 번째로는 각종 센서들을 사용하여 사용자의 졸음을



[그림 2] 뷰메이트



[그림 3] 마이카 버디 어플리케이션

인식하게 하는 방법이 존재한다.

주기적으로 알람을 울리는 방식으로 대표적으로 마이카 버디가 존재한다. 마이카 버디의 경우 특정 시간간격이 있으며 그 시간간격으로 지속적으로 알람이 울리기에 이 알람을 끄기 위해서 특정 행동을 취해야하므로 결국 졸지 못하게 하는 방식이다. 이 방식의 문제점은 정기적으로 알람은 울리지만 졸음 상태 자체를 깨게 하기는 힘들며 시속 100km의 경우 3초만 눈을 감아도 80m를 주행하는 고속도로 환경 특성상 알람 사이의 기간 동안 다시 졸음 운전의 위험이 있으며 주기가 짧을 경우 사용자의 어플리케이션의 운용 피로가 상당하다.

또 다른 방식으로 사람의 눈을 지속적으로 확인하는 뷰메이트라는 기기가 존재한다. 이 기기는 사람의 얼굴을 지속적으로 확인하기에

사람의 졸음 확인할 수 있는 장점이 있지만 얼굴을 확인하는 방식이기에 밤일 경우, 빛이 세게 비치는 경우, 안경을 쓰는 경우, 머리가 긴 경우등의 상황에서 인식이 힘들다는 단점이 있다. 센서를 부착하는 방식은 기기를 부착해야 하기에 추가적인 비용이 들며 아직은 시험 단계이다.

○ 프로젝트의 목적

졸음 운전이 교통 안전에 미치는 문제점이 크지만 기존의 졸음 운전 솔루션들은 졸음 상태를 크게 개선하지 못하거나 추가 비용이 들어 가는 등의 문제점이 있다. 이 문제점을 타파하기 위하여 이번 솔루션에서는 동승자에 초점을 맞추었다. 졸음 운전 사고의 62%의 경우 동승자가 존재하지 않았다는 결과가 있다. 법원의 판례에 따르면 졸음 운전의 동승자의 중요도가 30%가 있다고 보고 있다. 동승자의 존재가 졸음 운전의 사고 감소에 직접적인 영향을 주는 것이다.

이에 의하여 기존의 방식의 문제점을 해결하기 위하여 졸음 상태를 미리 감지하여서 그 상황에 마치 동승자가 존재하는 느낌을 주는 솔루션을 개발하려고 한다.

□ 프로젝트 개요

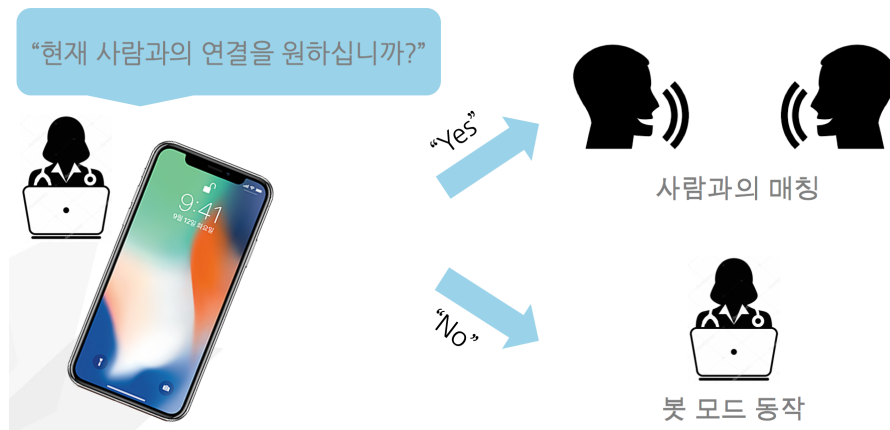


[그림 4] 공공 데이터 API

○ 프로젝트 소개

본 솔루션에서는 공공 데이터를 사용하여서 위험상황을 파악한다.

공공데이터는 공공 데이터 포털이나 한국도로 공사등의 open api를 사용한다. 사고 다발 지역이나 변화가 적은 직선 지역, 운전을 하는 시각, 운전을 하는 지역의 날씨 등의 요소들이 졸음 운전의 위험 상황이라고 할 수 있다. 이러한 졸음 운전의 위험상황들을 단순히 사용자가 통계 그래프를 확인하거나 임의로 취사 선택하는 것이 아니라 open api와 통계 프로그래밍 언어 R을 사용하여서 통계적인 기법을 사용하여서 산출해 낸다. 이 때 수집하는 open api 데이터는 사용자의 위치를 기반으로 수집한다. 중요한 것은 사용자가 해당 지역에 진입하기 전에 사용자가 미리 데이터를 수집하여서 위험 지역을 알아내는 것이 중요하다.

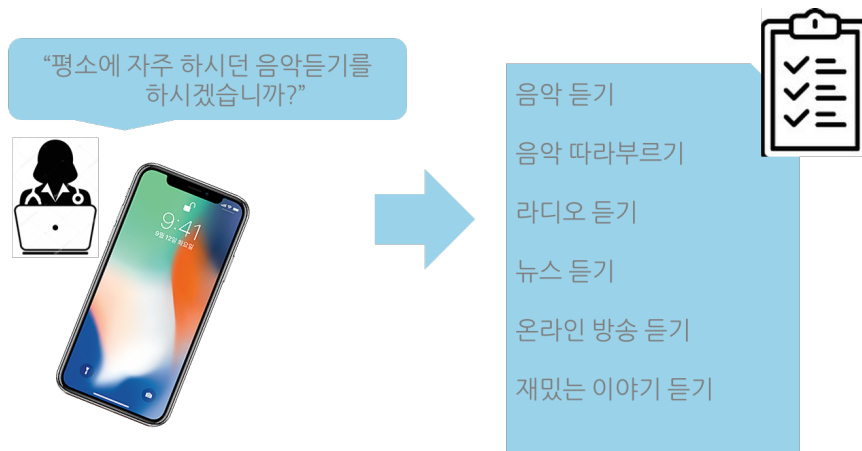


[그림 5] 봇모드와 사람과의 연결모드 선택

해당 위험 지역에 도달하였을 경우 서비스가 자동으로 동작하여서 졸리지 않게 하도록 한다. 이 때 서비스는 봇 서비스로서 동작한다. 봇은 보이스봇으로 음성으로 이야기하며 사용자에게 사람과 매칭을 시킬지 봇 모드를 계속해서 사용할지 물어본다. 매칭하는 사람은 이 솔루션을 사용하고 있는 다른 사용자를 의미한다. 이 사용자 역시 또 다른 장거리 운전자가 될 것이다. 서로 다른 운전자끼리 매칭이 됨으로서 운전자들 서로가 마치 동승자처럼 옆에서 있어서 말동무가 되어 줄 것이다.

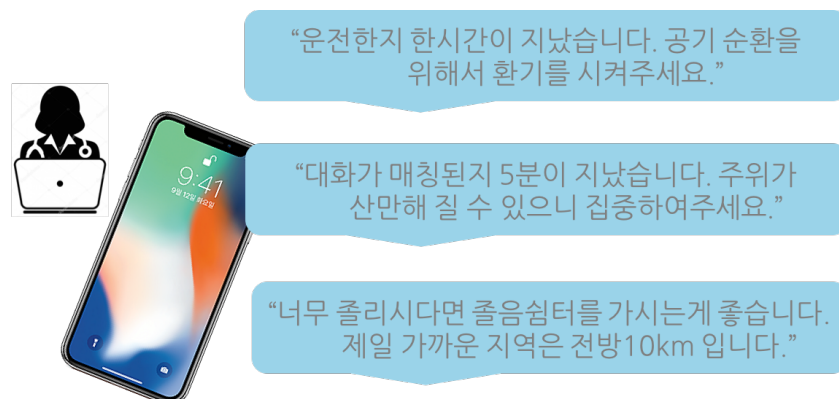
그러나 모든 사람들이 사람과의 매칭을 원하지는 않는다. 이 경우 봇모드로의 연결을 시켜준다. 봇이 마치 옆에서 동승자처럼 주위를

환기 시켜주는 것이다. 봇이 사용자가 졸리지 않도록 다른 작업들을 제시한다.



[그림 6] 봇모드 상태에서 작업 선택

봇모드에서 선택할 수 있는 작업들은 사용자에게 주위를 환기 시켜서 사용자가 졸리지 않게 하는 것이다. 이 작업은 자주 선택하거나 마지막으로 선택한 작업들을 우선적으로 물어본다. 그 작업을 선택하지 않는다면 봇모드에서 선택할 수 있는 작업들을 선택할 수 있다. 운전 중이기에 손을 사용할 수 없으므로 이 작업의 선택을 음성인식으로 처리한다.

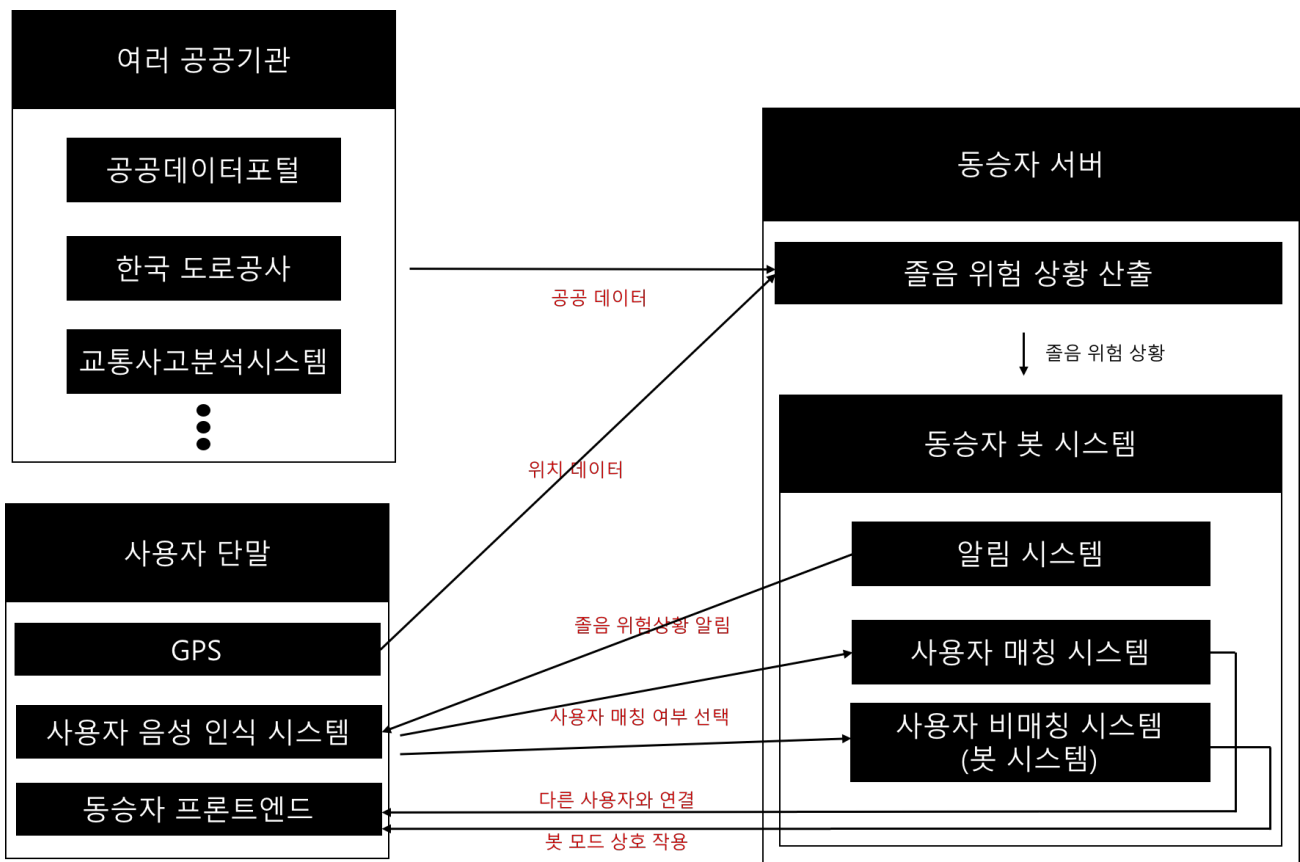


[그림 7] 주위를 계속 환기시키기 위한 솔루션

이 솔루션의 가장 중요한점 중 하나는 주위를 계속해서 환기시켜준다는 것이다. 옆에서 동승자가 계속 해서 말을 걸어주는 것처럼 솔루션

선이 동작하는 와중에도 사용자가 계속해서 주위를 환기시킬 수 있도록 작동하게 된다. 가령 대화가 매칭되었을 경우 사용자가 대화에 과몰입 하지 않도록 알람을 주거나 공기를 순환해서 졸음에 도움이 되게 하는 등의 역할을 하여서 사용자가 안전운전을 하며 졸음 운전을 하지 않게 하는데 지속적으로 관리를 하게 된다.

○ 시스템 구성도

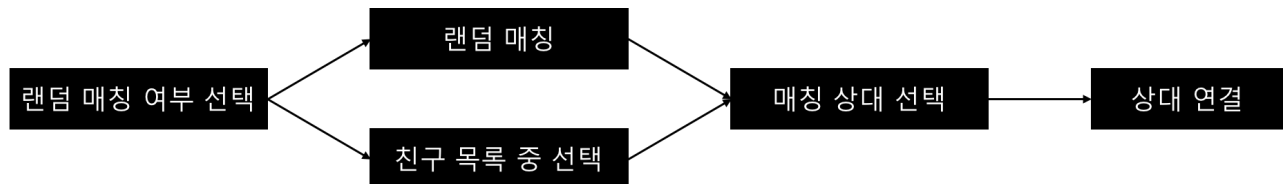


[그림 8] 전체 시스템 구성도

전체 시스템은 위와 같이 구성된다. 공공기관에서 공공데이터를 동승자 서버에 전달하고 사용자 단말에서(모바일)는 위치 정보를 동승자 서버에 전달한다. 이 전달된 데이터를 바탕으로 졸음 위험 상황을 산출하게된다.

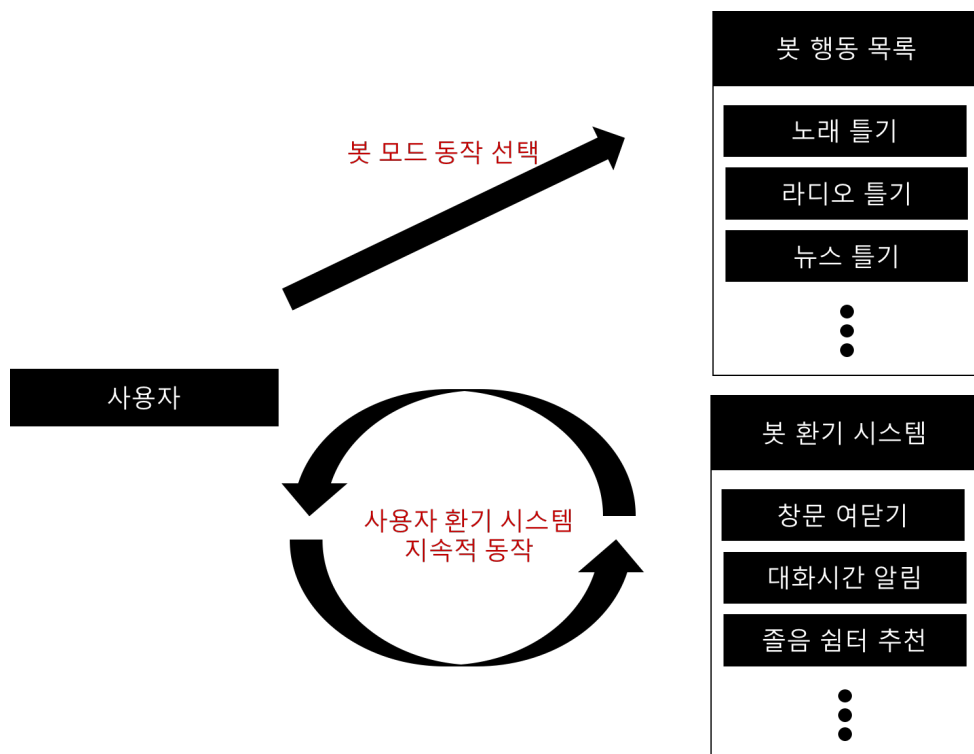
위의 과정에서 구해진 데이터를 바탕으로 동승자 봇 시스템을 동작시킨다. 졸음 위험상황과 그 후에 행해질 작동 모드를 선택한다. 작

동 모드는 총 2가지로 사용자 매칭 시스템과 사용자 비매칭 시스템을 둘 중 하나를 선택할 수 있다. 사용자 비매칭 시스템은 봇 시스템을 의미한다. 사용자 매칭 시스템을 선택할 시 다른 동승자 솔루션을 사용하는 사용자와 연결이 되며 봇모드를 선택할시 봇의 줄임 솔루션과 연결된다.



[그림 9] 사용자 매칭 시스템 순서도

사용자 매칭 시스템은 위와 같이 진행된다. 먼저 랜덤 매칭 여부를 선택한다. 그 다음 랜덤 매칭을 선택하였다면 현재 동승자 서버에 존재하는 사람들 중 랜덤으로 연결 시켜준다. 등록된 친구가 있다면 등록된 친구를 선택해서 매칭할 수 있다. 매칭 상대가 결정되면 상대와 연결되서 서로 대화를 나눌 수 있다.



[그림 10] 봇 시스템 동작 로직

사용자 비매칭 시스템인 봇 시스템을 선택할 경우 사용자는 봇 행동에 미리 정의된 것을 음성 인식을 통해 선택할 수 있다. 노래를 듣거나 라디오를 듣거나 뉴스를 들을 수 있으며 이러한 봇 행동들은 사용자가 자주 사용하는 봇 목록을 저장하게 되어서 먼저 사용자에게 자주 사용했던 목록을 묻고 그 행동을 우선적으로 실행할 수 있게 해주며 만약 사용하지 않는다면 다시 목록을 선택해서 사용할 수 있다. 이 와중에서 환기 시스템은 지속적으로 동작해서 사용자가 자칫 지루해지거나 아무 반응이 없는 상태에 익숙해져 졸음이 오는 상황을 방지하게 한다.

□ 프로젝트 수행 방법 및 일정

○ 주요 기능별 수행방법

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 웹크롤링을 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 졸음 운전에 미치는 비율을 산출 - 검정을 통해서 받아낸 데이터와 그 비율을 바탕으로 졸음 위험지역을 산출 - 비 위험지역과 위험지역의 차이를 R을 이용한 통계적 검정을 이용하여 산출
음성 인식 및 TTS	<ul style="list-style-type: none"> - 네이버 클로버를 이용해서 TTS를 사용하여 음성 서비스를 사용함 - 동승자 서버에서 받은 데이터를 TTS를 이용해서 사용자에게 상황 및 서비스를 알림 - 네이버 클로버를 이용해서 음성인식 서비스를 사용함 - 음성인식에 대한 오차를 줄이기 위해서 머신러닝을 통하여 학습하게 함 - 오차 학습에 대한 데이터는 nodejs의 웹크롤링을 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 확인하여서 차에 타고 있는 사람을 바탕으로 매칭시켜준다. - 랜덤 매칭을 끝낸 후 사용자를 친구등록 할 수 있다 - 친구 매칭을 선택할 경우 저장되어있는 친구 목록중에 한명을 선택해서 볼 수 있다.

사용자 비매칭 시스템 (봇 시스템)	<ul style="list-style-type: none"> - 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)	<ul style="list-style-type: none"> - 봇은 백그라운드에서 계속하여서 사용자가 운전중인 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)	<ul style="list-style-type: none"> - 안드로이드와 ios에서 모두 개발하도록함 - 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 주고 받는 데이터는 restful디자인으로 주고받음
동승자 서버	<ul style="list-style-type: none"> - amazon ec2 서버를 사용 - 서버의 언어는 noedjs와 python중 추후에 더 적합한 언어로 사용 - 사용자의 중요한 데이터는 RDBMS이면서 오픈소스인 mariaDB를 사용 - 실시간으로 처리해야할 데이터들은 NoSQL인 HBase와 인메모리DB인 redis를 사용하도록 함 - 동승자 서버에 사용자 매칭 시스템과 봇 시스템이 모두 들어 있음 - 동승자 서버와 단말과의 데이터 전송은 restful로 주고 받음
UI	<ul style="list-style-type: none"> - 어플리케이션은 졸음 방지라는 특수한 상황 상 손을 거의 쓰지않음 - 어플리케이션은 친구목록을 확인하거나 설정을 바꾸는 등의 간단한 기능만 사용할 수있음 - 어플리케이션에서 동승자 모드에서 자동으로 봇모드와 랜덤 매칭 모드를 선택할 수있게 하는등의 선택을 할 수 있음 - UI의 개발은 sketch를 사용하도록 함

○ 적용기술

구분		항목	세부 내용
S/W 개발환경	OS	MacOS Android Linux(Ubuntu)	기본적인 개발은 MacOS에서 개발하되 테스트를 위하여 Android 환경을 사용할 수 있고 서버 테스트를 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

○ 추진 일정

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

□ 기대효과 및 활용분야

<p>사용자 측면</p>	<ul style="list-style-type: none"> - 사용자는 항상 졸음 운전의 위험에 처해있고 본인의 의지로 졸음 상태에서는 벗어나기 힘들지만 졸음 방지 솔루션인 동승자 어플리케이션을 이용하여서 졸음 상태에 돌입하지 않고 안전한 운전을 하도록 도움을 줌 - 기존의 졸음 운전 해결책들은 명확하지 않거나 가격이 많이 드는데 이 솔루션을 사용함으로써 가격 부담을 줄이면서도 기존의 해결책 보다 좋은 효과를 누릴 수 있음
<p>비즈니스 측면</p>	<ul style="list-style-type: none"> - 현재 여러 자동차 업체들이 자율 주행 다음으로 주시 하는 것이 졸음 운전 대책 서비스이므로 여러 자동차업체들과 협력을 맺을 수 있음 - 자동차와 소셜 네트워크가 결합된 형태로 카카오톡, 라인과 같이 유저풀을 확대하여서 그 유저풀을 바탕으로 추가적인 비즈니스를 행할 수 있음 - 추후 해당 솔루션을 네비게이션이나 아이트래킹을 도입한 IOT등에 접목하여서 상품을 팔아서 이득을 취할 수 있음
<p>개발자 측면</p>	<ul style="list-style-type: none"> - 동승자 행동들을 다양하게 추가함으로써 확장성을 확보할 수 있음, 추후에 새로운 솔루션 추가시에 손쉽게 추가가 가능하고 api를 제공하여서 다른 개발자의 참여를 유도할 수 있음 - 소셜 네트워크가 결합된 형태를 이용하여서 이를 바탕으로 새로운 솔루션을 개발하여서 추가하게 할 수 있는게 간편하며 api를 제공하여서 다른 개발자의 참여를 유도할 수 있음