

# 2022년도 SW마에스트로 과정 제13기 프로젝트 중간점검 보고서

## 1. 프로젝트개요

프로젝트명	지금갈게 (Where To Drink)		
프로젝트 요약	실시간으로 방문 가능한 주점을 찾아 예약해주는 소프트웨어 플랫폼		
기술 키워드	#예약플랫폼 #실시간예약 #Pub/Sub #WebSocket #UI/UX		
ICT연구개발 기술 분류	① SW·AI - ② 응용SW		
팀 명	삼시세끼	팀 원	백준호, 최지윤, 한재혁
목적 및 필요성	술자리를 잡을 때 일일이 돌아다니거나 주점마다 전화를 걸어 자리가 있는지 확인하는 경우가 대다수. 특히, 2차/3차와 같이 즉흥적으로 모임 자리가 이어지는 경우 미리 시간과 인원을 특정하여 예약하는 것이 불가능. 실시간 예약 서비스를 구현하여 자리 문의 및 예약 문제를 해결하고, 주점업자의 편리한 예약관리를 도움.		
프로젝트 개요	"저희 5명이고 20분 뒤 갈 것 같은데 거기 자리 있을까요?" 번화가에서 손쉽게 단체예약을 진행하고, 바로바로 이용 가능한 주점을 찾기 위한 주점 이용자와 손님들의 예약목록들을 손쉽게 관리하고 높은 테이블 회전율을 지향하는 주점업자들을 연결해줄 수 있는 실시간 예약 서비스		
수행 방법 및 프로젝트 관리	<p>&lt;수행 방법&gt;</p> <ol style="list-style-type: none"> <li>1) 실시간 자리 예약 알고리즘 개발 및 예약 서비스 개발</li> <li>2) 주점업자 실시간 예약 목록 확인 및 관리 페이지 개발</li> <li>3) 노쇼 방지 거버넌스 설계 및 구현</li> <li>4) 실시간 예약에 알맞은 UI/UX 디자인(주변 검색, 자리 검색, 메뉴 검색, 기타)</li> </ol> <p>&lt;프로젝트 관리&gt;</p> <ul style="list-style-type: none"> <li>- 애자일 방법론(스크럼) 적용</li> <li>- 매일 아침 스크럼 미팅을 통한 팀원 간 현황 공유 및 이슈 처리</li> <li>- Github Organization을 사용해 프로젝트 형상관리 및 코드리뷰</li> </ul>		
개발 진행현황	<ul style="list-style-type: none"> <li>- 실시간 자리 확인 : 구현 완료</li> <li>- 실시간 예약 : 구현 완료</li> <li>- 주점업자 예약 관리 : 예약자 위치확인 향후 개발 필요</li> <li>- 주점이용자 예약 관리 : 길찾기 기능 향후 개발 필요</li> </ul>		
결과물 형태 및 서비스 방식	<p>최종 결과물 : 웹 서비스 제공을 위한 전용 웹 사이트와 스마트폰 애플리케이션</p> <p>업로드 방식 :</p> <ul style="list-style-type: none"> <li>- 웹 페이지 URL</li> <li>- 플레이스토어, 앱스토어 업로드</li> <li>- 시연 영상</li> </ul>		
기대 효과 및 활용 방안	<ul style="list-style-type: none"> <li>- 주점 이용자 : 원하는 식당을 찾는 비용 감소</li> <li>- 주점업자 : 원활한 예약 관리 및 매장 운영</li> <li>- 비즈니스 : 코로나로 침체된 외식업 상권 활성화 기대</li> <li>- 개발자 : 실시간성, 위치 기반 서비스 개발 역량 상승</li> </ul>		

## 2. 프로젝트 중간보고서

### □ 목적 및 필요성

#### ○ 문제인식

1. 주점 이용 시 일일이 돌아다니거나 주점마다 전화를 걸어 자리가 있는지 확인하는 경우가 대다수이다. 특히, 2차/3차와 같이 갑작스럽게 이루어지는 모임의 경우 미리 시간과 인원을 특정하여 예약하는 것이 불가능하다.
2. 대학가 주점 사장님들과의 인터뷰 결과, 다음과 같은 어려움을 발견했다.
  - 영업 중간에 자리 문의 및 예약 전화를 받기 힘들다.
  - 예약에 대한 기록이 수기로 작성되기 때문에 예약 관리에 어려움이 있다.
  - 손님들의 중복 예약으로 인한 노쇼가 발생하며, 피해는 사장님이 떠안는다.
3. 대형 포털사에서 제공하는 예약 서비스는 최소 하루 전 예약 요청이 필수이며 당일 예약 및 취소가 불가능하다.


#### ○ 기획 의도(문제해결)

주점업 대상 실시간 예약 서비스를 제작하여 오프라인으로 이루어지는 자리 문의 및 예약의 비효율성을 해결한다. 이에 노쇼 방지 정책도 도입하여 노쇼로 인한 주점업자의 손실을 최소화하고자 한다.

#### ○ 시장/소비자 동향/분석, 경쟁 제품 비교/차별화 요소

<그림 1> 식품산업통계정보시스템에 따르면 올해 3월 위드코로나 시행 이후 한 달 전과 비교해 요식업종 특히, 주점업종의 매출액이 유흥주점 62.1%, 맥주 전문점 66.3% 등 폭발적으로 늘었다. 단체 행사나 모임이 증가해 주점 시장에 활기를 되찾고 있는 분위기이며 단체 예약에 대한 수요가 증가하고 있다.

한식 일반	23.5%
한식 면	25.2%
한식 육류	25.6%
일식	19.6%
유흥주점	62.1%
맥주 전문점	66.3%
커피 전문점	15.1%



※자료: 한국농수산식품유통공사

<그림1> 요식업종 2월 대비 3월 매출 증가량

<표 1> 통계청의 ‘2021년 외식업체 조사결과’에 따르면 외식 업체 중 주점업의 비율이 2021년 기준 15.8%로 한식 다음으로 가장 높은 것으로 나타났다.

구분	2018	2019	2020	2021
한식	45.6	44.9	44.2	43.6
중식	3.5	3.6	3.5	3.5
일식	1.5	1.7	1.9	1.9
서양식	1.7	1.7	1.8	1.9
기타 외국식	0.4	0.5	0.6	0.7
기관 구내식당업	1.6	1.6	1.6	1.5
출장/이동 음식점업	0.1	0.1	0.1	0.1
제과점	2.7	2.5	2.7	3
피자/햄버거/샌드위치 및 유사 음식점업	2.4	2.6	2.7	2.8
치킨 전문점	4.9	5.5	5.2	5.2
김밥 및 기타 간이 음식점업	7.3	6.1	6.1	6.1
간이 음식 포장 판매 전문점		1.1	1	1
주점업	18.2	17.5	16.8	15.8
비알코올음료점업	10.1	10.7	11.8	12.9

<표1> 외식업체 업종별 비중 (단위: %)

유사 서비스로는 ‘네이버 예약’과 ‘캐치테이블’ 등이 있다. 이들은 주점을 포함하여 다양한 음식점에 방문 예약을 하고, 해당 예약을 관리할 수 있다는 점에서 ‘지금갈게’의 경쟁 상품이라고 할 수 있다. 아래 표는 ‘지금갈게’와 타 경쟁 서비스를 비교 분석한 내용이다.

구분	지금갈게	네이버 예약	캐치테이블
주요 고객 업종	외식업 중 주점업	다양한 업종	요식업종
실시간 자리 확인/예약	O	X	X
노쇼 관리 방법	신용 등급 제도, 방문 유도 푸시알림	선결제 후 수수료	보증금 납부
방문 시간 지정 예약	실시간 우선(확장 가능)	핵심 기능(한계)	핵심 기능(한계)
특징	노쇼 관리 가능, 소/대규모 주점업 모두 적합	노쇼 관리 부재, 당일 예약 불가능	고급 레스토랑에 적합

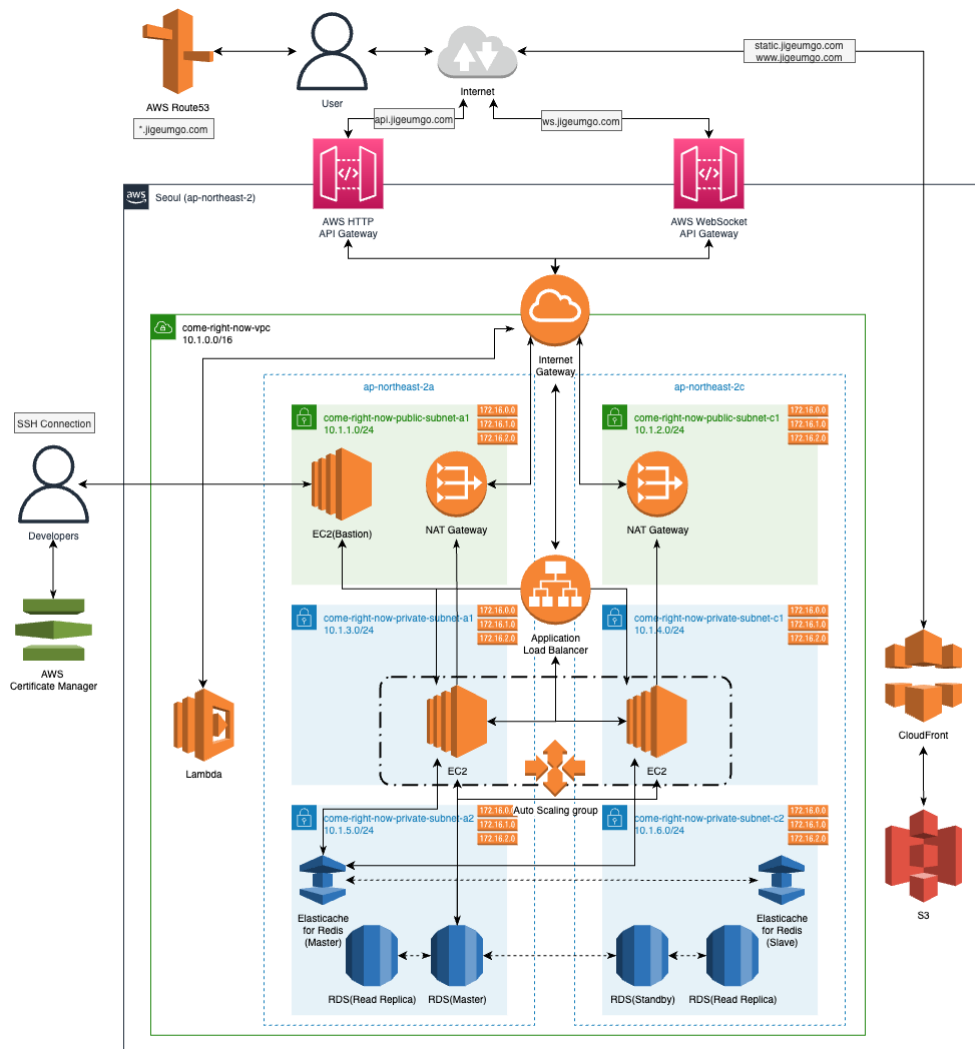
‘지금갈게’와 타 서비스의 가장 큰 차별 점은 실시간 자리확인 및 예약 기능이다. ‘네이버 예약’과 ‘캐치테이블’은 구체적인 시간을 특정하여 예약하는 방식으로 동작한다. ‘지금갈게’와 달리 지금 당장 자리가 있는 곳을 찾아 방문 직전 예약할 수 있는 기능은 제공하지 않는다. 주로 날짜 및 시간, 인원을 특정하여 방문 전 예약이 이루어지는 일반 음식점과 달리, 주점업의 경우 즉흥적으로 방문 직전 자리를 탐색해 예약이 이루어지는 경우가 많으므로 기존의 예약 서비스를 주점업 예약에 적용하기에는 적합하지 않다.

## □ 프로젝트 개요

### ○ 프로젝트 소개

주점 이용자(User)는 모바일을 통해 인원의 수를 설정하고 주변의 영업 중인 주점에 당장이나 짧은 시간 안에 갈 곳을 찾고 있다는 알림을 보낸다. 알림을 받은 주점업자는 이용자의 규모와 도착 가능 시간, 실시간 매장의 상태 등을 확인해 수락/거절 응답을 보낸다. 주점 이용자에게는 수락한 주점들의 정보가 나열되고, 하나를 선택해 예약을 체결한 뒤 해당 주점을 이용하게 된다.

### ○ 시스템 구성도



- AWS Route53을 통한 서비스 도메인 라우팅
- S3로 정적 웹사이트 호스팅 및 정적 파일 저장, CloudFront를 통한 웹사이트 캐싱
- Rest API, Websocket API Gateway로 통신 프로토콜 분리

- VPC로 격리된 가상 클라우드 네트워크 구성
- 퍼블릭 서브넷 2개, 프라이빗 서브넷 4개로 구성
- 개발자 서버 접속을 위한 Bastion Host, NAT Gateway로 구성된 퍼블릭 서브넷 2개 구성
- AWS Certificate Manager로 백엔드 서버 접속을 위한 인증서 관리
- 백엔드 서버로 구성된 프라이빗 서브넷 2개 구성
- AWS Application Load Balancer와 함께 부하 분산을 위해 이중화 구성
- 백엔드 서버 Auto Scaling
- DB 프라이빗 서브넷 2개 구성
- PUB/SUB, 데이터 캐싱을 위한 Elasticache, Master-slave 이중화 구성
- AWS RDS 이중화 구성, 읽기전용(Read-Replica) 구성
- AWS Lambda를 통한 추가 API 구성

#### ○ 주요 기능

- 실시간 자리 확인 및 예약: 주점 이용자와 주변 주점의 실시간 통신으로 주점 이용자는 수용 가능한 주점을 확인할 수 있다. 방문 가능한 주점의 위치, 대표 메뉴 등 정보를 확인할 수 있으며 그 중 원하는 곳을 예약하고 방문할 수 있다.
- 주점업자 실시간 예약목록 확인하기: 현재 체결된 예약들에 대한 목록들을 열거하여 주점업자에게 보여준다.

#### ○ 개발 환경

구분		상세내용
S/W 개발환경	OS	[macOS/Windows] 개발 환경 (웹, 모바일 애플리케이션)
		[Linux] 서버 인프라(컴퓨팅, 스토리지, 데이터베이스 등)의 배포
	개발환경(IDE)	[VS Code] 백엔드(Node.js), 프론트엔드(React) 개발 및 배포
	개발도구	[Postman] 백엔드 API Integration 테스트
		[Figma] 디자인 페이지 관리 및 제작
		[AWS RDS] 개발용 데이터베이스
		[MySQL Workbench] 데이터베이스 관리
	개발언어	[TypeScript] 정적타입을 사용하여 가독성과 활용도 높은 코드 작성
		[JavaScript] 백엔드(Node.js), 프론트엔드 브라우저 로직 제어
H/W 구성장비	디바이스	[아이폰 12미니] 서비스 클라이언트(주점이용자) 테스트
		[아이패드 프로 4세대] 서비스 클라이언트(주점업자) 테스트
프로젝트 관리환경	형상관리	[Github Organization] 프론트엔드, 백엔드 코드 버전관리 및 코드리뷰 환경 구성
	의사소통관리	[Jira] 스크럼 개발방법론 적용 및 프로젝트 진행도 관리
		[Notion] 프로젝트 개발, 기획, 디자인 문서 및 회의록 작성
	기타사항	[Webex] 디자이너, 멘토, 팀 간 원격 소통

## □ 수행 방법 및 프로젝트 관리

### ○ 주요 기능별 수행방법

실시간 자리 확인	<ul style="list-style-type: none"> <li>- 주점 이용자는 인원수를 설정하여 탐색 요청을 보낸다.</li> <li>- 지금갈게 시스템은 주변 주점업자들에게 해당 요청을 알림 형태로 전달한다.</li> <li>- 요청을 받은 주점업자는 수용 가능 여부를 판단하여 수락/거절 응답을 보낸다.</li> <li>- 시스템은 수락 응답을 한 주점의 목록을 주점 이용자에게 실시간 전달한다.</li> <li>- 주점 이용자는 수락 응답을 한 주점에 대하여 주점 위치 및 정보를 확인할 수 있다.</li> </ul>
실시간 예약	<ul style="list-style-type: none"> <li>- 주점 이용자는 나열된 주점 중 한 곳을 선택해 예약을 체결한다.</li> <li>- 시스템은 선택된 주점업자에게 예약 체결 알림을 보낸다.</li> </ul>
자영업자 예약 관리	<ul style="list-style-type: none"> <li>- 시스템은 주점업자에게 현재 체결되어있는 예약 리스트를 실시간으로 보여준다.</li> <li>- 주점업자는 예약 리스트에서 방문 인원 수, 방문 시간, 예약자 전화번호 등 예약정보를 확인할 수 있다.</li> <li>- 주점업자는 특정 예약 건에 대해 취소 사유 선택과 함께 취소 처리가 가능하다.</li> <li>- 예약이 취소되면 시스템은 주점 이용자에게 예약 취소 알림을 보낸다.</li> <li>- 예약자가 예약 후 실제로 방문했다면 주점업자는 예약 상태를 종결한다는 의미로 체크인 처리한다.</li> <li>- 체크인 처리가 되면 시스템은 주점 이용자에게 체크인 알림을 보낸다.</li> </ul>
주점 이용자 예약 관리	<ul style="list-style-type: none"> <li>- 주점 이용자의 체결된 예약에 대해 시스템은 메인화면에 현재 체결되어 있는 예약 내역을 보여준다.</li> <li>- 주점 이용자는 출발 시간과 도착 시간 안내 등 상세 예약 정보와 주점 정보를 확인할 수 있다.</li> <li>- 주점 이용자는 일정 시간동안 체결된 예약에 대해 취소 요청을 보낼 수 있다.</li> <li>- 주점 이용자는 예상 도착 시간 안에 도착하지 못할 것으로 판단될 경우 도착 지연 신청을 할 수 있다.</li> <li>- 주점 이용자가 예약을 취소하거나 도착 지연 신청을 하면 시스템은 주점업자에게 즉시 알림을 보낸다.</li> <li>- 주점 이용자는 외부 길찾기 서비스에 연결되어 예약 주점까지의 경로를 확인할 수 있다.</li> </ul>

○ 수행방법 확보방안(아웃소싱 등)

- 내부 개발: 프로젝트 요구사항을 팀원 간 합의하고 이에 맞는 프로젝트 설계, 개발을 진행한다. 개발 사이에 프로젝트 코드에 대한 단위테스트와 통합테스트를 진행하고, 스프린트 백로그 단위로 프론트엔드-백엔드 E2E 테스트를 진행한다.

- 아웃소싱 개발: 실시간 주점 검색 및 예약에 특화된 UI/UX 개발을 위한 전문가를 의뢰한다. 상세한 개발 일정 및 요구사항을 전달하고, 산출물이 부분적으로 나올 때마다 리뷰 후 디자인을 프로젝트에 적용한다.

○ 추진 일정

기능명	구분	추진 내용	진척도	추진 일정					
				6월	7월	8월	9월	10월	11월
프로젝트 기반작업 (개발환경 구성 및 설계)	계획	프로젝트 로드맵 작성	100%						
	분석	스프린트 백로그 작성	100%						
		프로젝트에 적합한 기술스택 선정	100%						
	설계	시스템 아키텍처 설계	100%						
		UI/UX 와이어프레임 제작	100%						
	종료	노션 문서 정리	100%						
실시간 예약기능	계획	지라 백로그 하위 이슈 작성	100%						
	분석	요구사항 분석 및 기능명세 작성	100%						
	설계	API 문서 작성 및 디자인 페이지 제작	100%						
	개발	실시간 예약 (프론트엔드) 개발 및 실시간 예약 (백엔드) 개발	80%						
	테스트	Unit Test, Integration Test, E2E Test	80%						
	종료	지라 에픽 단위 데모 리뷰 및 회고	70%						
예약관리 기능	계획	지라 백로그 하위 이슈 작성	30%						
	분석	요구사항 분석 및 기능명세 작성	30%						
	설계	API 문서 작성 및 디자인 페이지 제작	30%						
	개발	예약 관리 (프론트엔드) 개발 및 실시간 예약 (백엔드) 개발	30%						
	테스트	Unit Test, Integration Test, E2E Test	20%						
	종료	지라 에픽 단위 데모 리뷰 및 회고	20%						
계정관리 기능	계획	지라 백로그 하위 이슈 작성	10%						
	분석	요구사항 분석 및 기능명세 작성	10%						
	설계	API 문서 작성 및 디자인 페이지 제작	50%						
	개발	계정관리 (프론트엔드) 개발 및 계정관리 (백엔드) 개발	10%						
	테스트	Unit Test, Integration Test,	10%						



기능명	구분	추진 내용	진척도	추진 일정					
				6월	7월	8월	9월	10월	11월
예약내역 (history) 및 통계 기능		E2E Test							
	종료	지라 에픽 단위 데모 리뷰 및 회고	10%						
	계획	지라 백로그 하위 이슈 작성	00%						
	분석	요구사항 분석 및 기능명세 작성	00%						
	설계	API 문서 작성 및 디자인 페이지 제작	50%						
	개발	예약내역&통계 (프론트엔드) 개발 및 예약내역&통계 (백엔드) 개발	00%						
	테스트	Unit Test, Integration Test, E2E Test	00%						
마이 페이지 기능	종료	지라 에픽 단위 데모 리뷰 및 회고	00%						
	계획	지라 백로그 하위 이슈 작성	00%						
	분석	요구사항 분석 및 기능명세 작성	00%						
	설계	API 문서 작성 및 디자인 페이지 제작	50%						
	개발	마이페이지 (프론트엔드) 개발 및 마이페이지 (백엔드) 개발	00%						
	테스트	Unit Test, Integration Test, E2E Test	00%						
	종료	지라 에픽 단위 데모 리뷰 및 회고	00%						

## ○ 역할 분담

담당		역할 및 상세활동
연수생	백준호	- 프로젝트 관리 - WebSocket 통신에 적합한 Scalable 아키텍처 설계 - 백엔드 개발
	최지윤	- DB모델링 - Redis-Pub/Sub 기반의 실시간 게시/구독 알고리즘에 적합한 백엔드 개발
	한재혁	- React 기반의 실시간 예약 프론트엔드 개발 - 아웃소싱 업체 관리 및 협업
멘토	강상진	- 프로젝트 기획 및 프로세스 멘토링 - 시스템 아키텍처 설계 멘토링
	박주형	- 개발환경 구성 및 기획 멘토링 - CI/CD환경 및 DEVOPS 멘토링
	정원용	- 웹 서비스 개발 멘토링 - 코드 리뷰 및 설계 디자인패턴 멘토링

○ 기술 습득 노력

담 당		기술 습득 노력
연수생	백준호	<ul style="list-style-type: none"> <li>- 인터넷 강의를 통한 Jira 사용법 습득, 애자일 방법론 프로젝트 적용을 통한 프로젝트 프로세스 이해</li> <li>- NestJS, TypeORM 공식문서 기반 학습을 통한 생산성 있는 백엔드 서버 개발</li> <li>- Websocket 소모임을 통한 양방향 통신 지식 공유 및 스터디</li> <li>- AWS 아키텍처 멘토링 및 AWS 문서를 통한 확장성 있고 안정적인 시스템 아키텍처 설계</li> </ul>
	최지윤	<ul style="list-style-type: none"> <li>- 인프런 등 온라인 NestJS, TypeORM 튜토리얼 강의를 수강하며 미니 프로젝트 실습을 통해 학습</li> <li>- Websocket 관련 공식 문서와 튜토리얼 강의를 통해 동작 원리와 프로젝트 적용법 습득</li> <li>- 프로젝트 진행 전 간단한 채팅 어플을 구현하며 WebSocket과 Nest.js 연동 및 문법 학습</li> <li>- 정기 멘토링 및 AWS 특강을 통해 올바른 시스템 아키텍처 설계 안목 증진 및 실습</li> </ul>
	한재혁	<ul style="list-style-type: none"> <li>- 프로젝트 진행 전 스터디를 통해 React의 동작원리를 학습하고 간단한 미니 프로젝트를 통해 숙련도 향상</li> <li>- 온라인 강의를 통해 TypeScript 지식 습득 및 공식 문서를 통해 TypeScript를 이용한 React 프로젝트 생성 및 개발에 적용</li> <li>- 구글링과 공식 문서를 통해 ReactDOM, Zustand, SocketIO, StyledComponents, SweetAlert2와 같은 각종 라이브러리의 사용법을 익히고 Notion과 같은 개인 메모 프로그램을 통해 정리하여 사용</li> <li>- 정기 멘토링 및 AWS 특강을 통해 기본적인 AWS 지식과 아키텍처 설계 지식 습득</li> </ul>

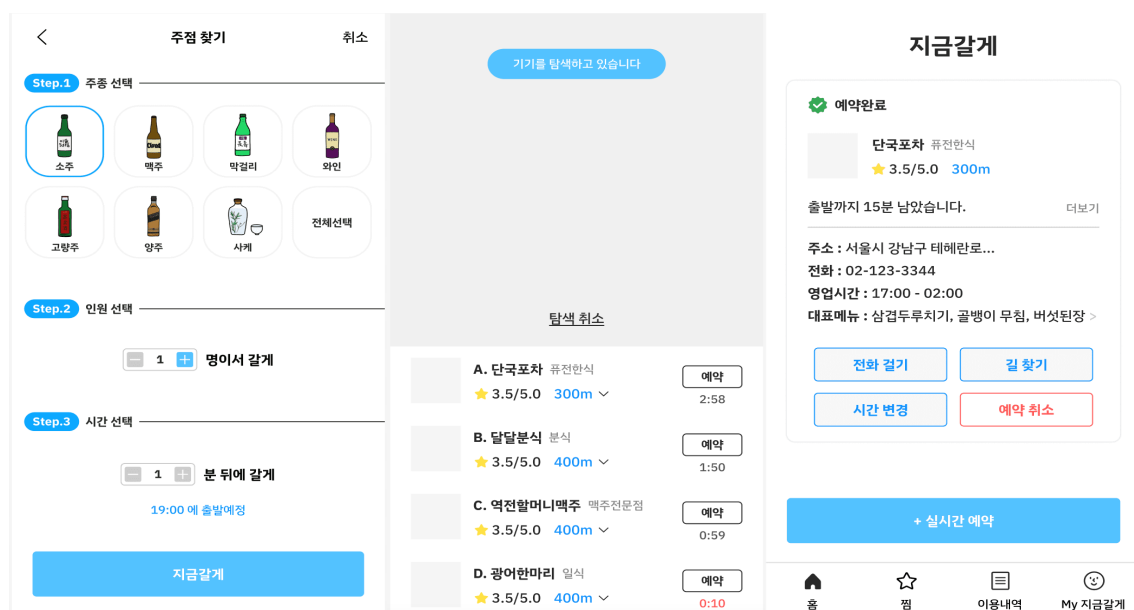
○ 문제점 및 해결방안

구 분	문제점 및 해결방안
관리 측면	<ul style="list-style-type: none"> <li>- 스크럼 방법론 적용 시 현황 공유 및 일정 스케줄링의 어려움 : 매일 아침 팀원과 스크럼 미팅을 통해 각자의 현황 공유 및 이슈관리</li> <li>- 스프린트 진행 도중 버그 해결, 리팩토링, 회의 후 수정본 반영의 어려움 : 스프린트 진행 도중에는 최대한 백로그 추가를 지양하고, 로드맵에 백로그를 추가하여 스프린트 종료 이후 우선순위에 대한 논의 후 스프린트에 추가</li> <li>- 주점업자 입장 반영의 한계 : 주점업자와 지속적인 인터뷰를 통해 Persona 형태의 니즈를 파악한 후 지속해서 서비스에 반영</li> </ul>
개발 측면	<ul style="list-style-type: none"> <li>- Stateful한 Websocket 서버 특성상 확장성(scale-out) 있는 서비스 구조 설계의 어려움 : Redis-Pub/Sub 기능을 활용하여 이벤트 통신 처리. socket.io redis-adapter를 통해 여러 서버에 연결하여 서비스 확장</li> <li>- 주점이용자와 주점업자간 양방향 통신을 할 때 서비스에 접속한 주점에 대하여만 이벤트 통신이 가능해야함 : onlineMap이라는 인메모리 객체를 만들어 소켓 접속시 이용자DB id와 소켓id를 저장함으로써 현재 접속중인 유저 현황 유지</li> <li>- Websocket Event 로그 라이브러리의 부재 및 테스트의 어려움 : NestJS의 Logger모듈을 상속받아 Websocket Event Logger모듈 제작, Postman-websocket request를 사용하여 user, store Websocket 이벤트 테스트 구성</li> </ul>

## □ 개발 진행현황

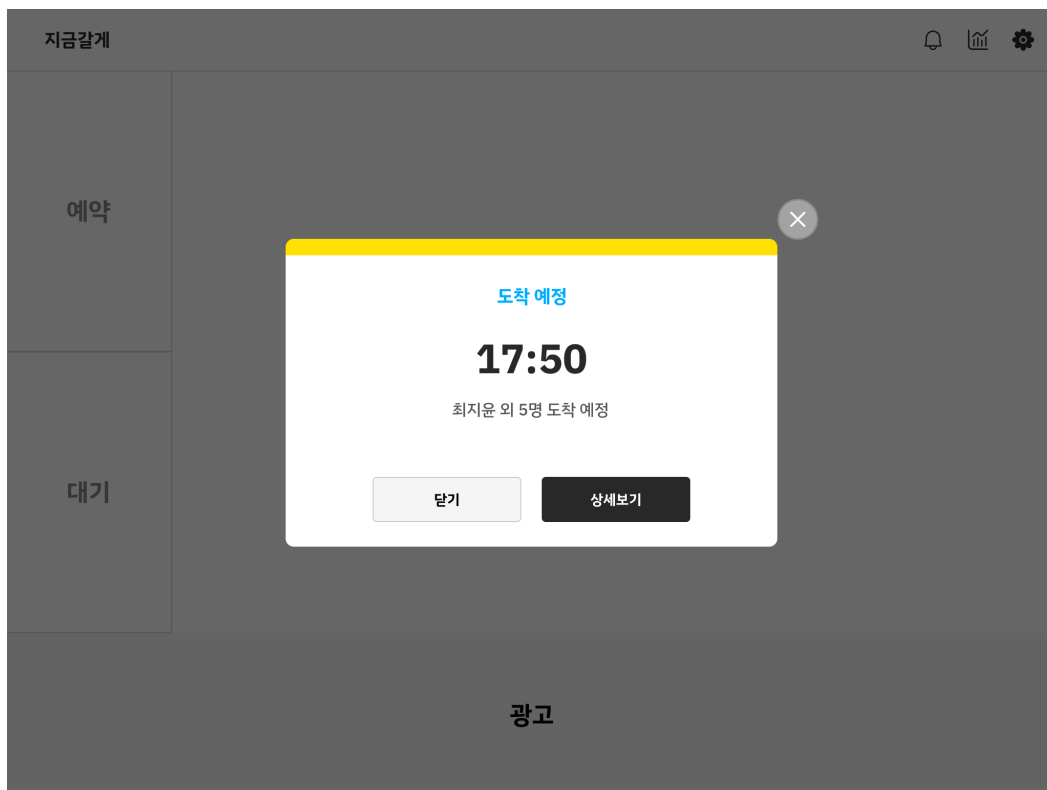
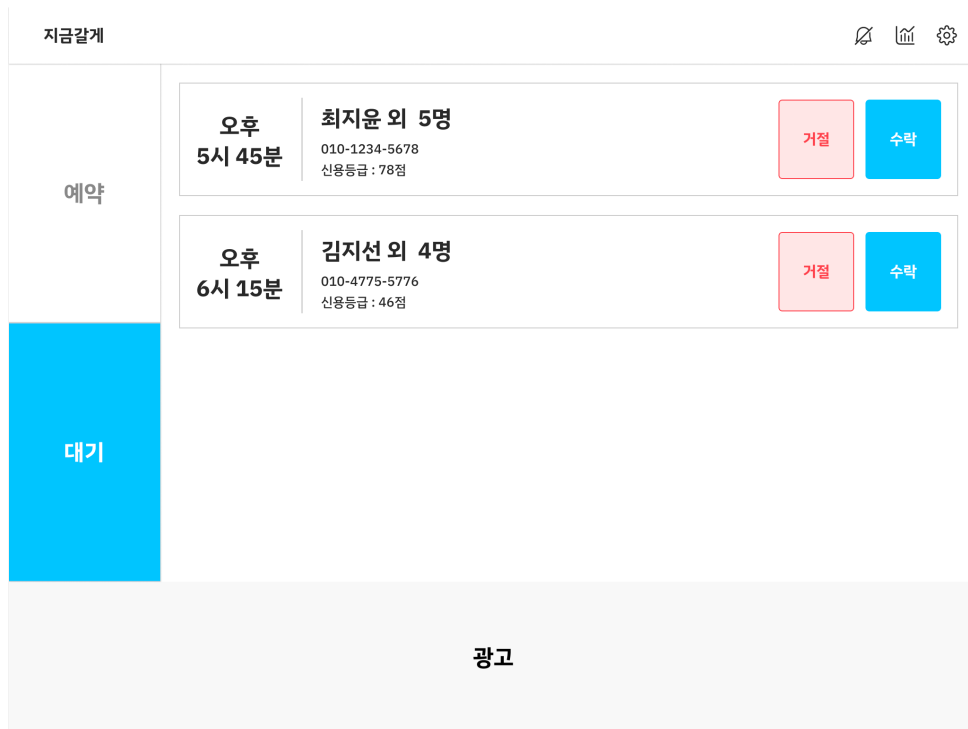
### ○ 프로젝트 주요 결과물

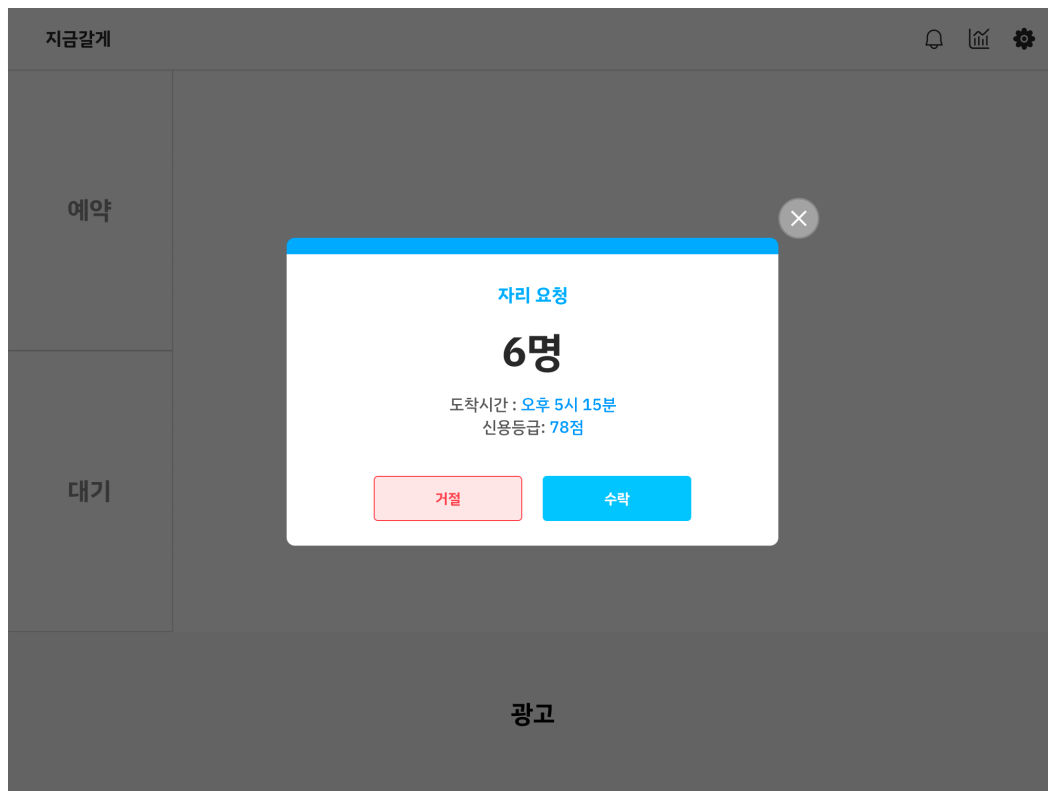
- 지금갈게 클라이언트
- 이용자가 사용하는 화면
- 근처 방문 가능한 주점 탐색 및 예약



- 지금갈게 사장님

- 스토어(사장님)이 사용하는 화면
- 예약 현황, 자리요청 현황 확인 및 예약 관리





○ 프로젝트 세부 산출물(개발물, 제작물)

## 1. 실시간 자리 탐색 기능

- 주점이용자

<화면 1> 주점이용자는 주종, 인원 수, 출발 지연 시간을 선택한 후 주점을 찾는다. 주점이용자의 현재 위치를 기반으로 반경 500m 내에 위치한 주점에 대해서 자리 확인 요청이 보내진다.

<화면 2> 주점이용자는 요청 보낸 주점업자 측의 응답을 기다린다.

<
주점 찾기
취소

---

**Step.1** 주종 선택

 소주	 맥주	 막걸리	 와인
 고랑주	 앙주	 사케	전체선택

**Step.2** 인원 선택

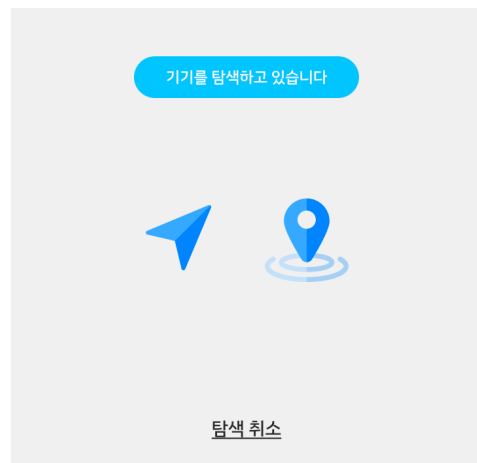
명이서 갈게

**Step.3** 시간 선택

분 뒤에 갈게

19:00 에 출발예정

지금갈게

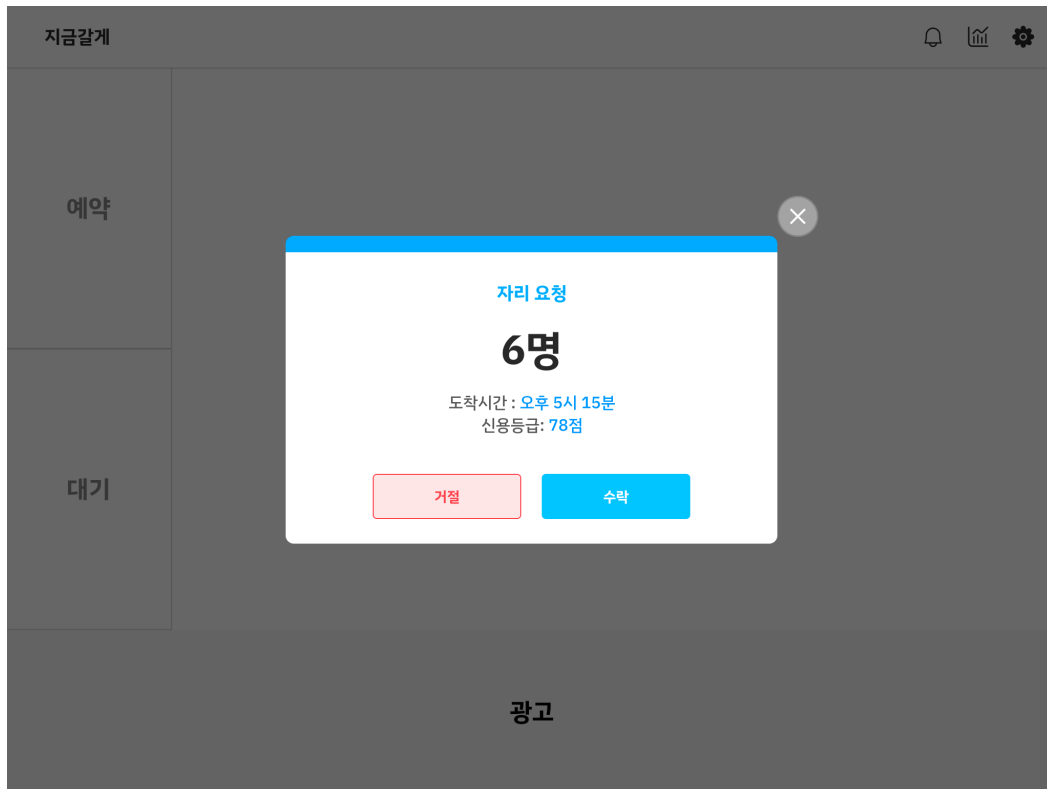


애니메이션

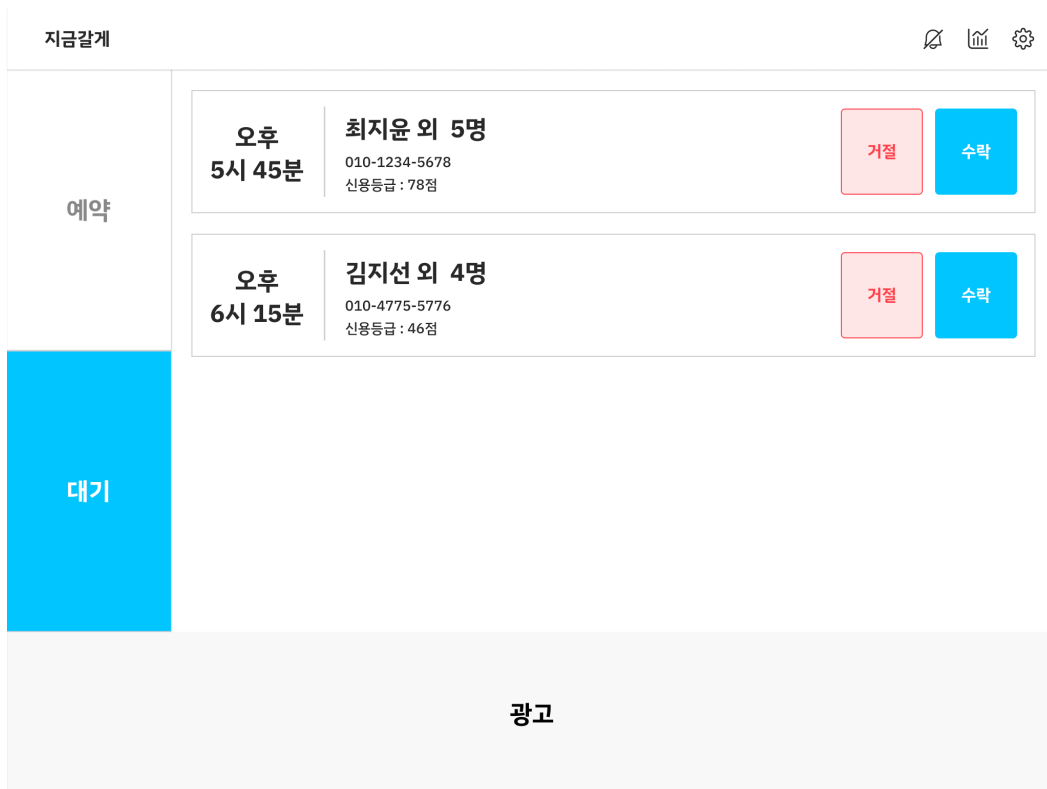
- 주점업자

〈화면 3〉 주점업자는 주점이용자 측에서 보낸 자리 확인 요청에 대해 실시간으로 알림 팝업을 받는다. 주점업자는 매장의 현 상황을 살핀 후 요청받은 인원수와 도착 예상 시간을 고려하여 수락 또는 거절 응답을 보낸다.

〈화면 4〉 주점업자가 즉시 팝업으로 응답하지 못한 요청들은 ‘대기’ 메뉴에서 확인할 수 있다. 마찬가지로 주점업자는 수락 또는 거절 응답을 할 수 있지만, 각각의 요청들은 10분 동안만 유효하기 때문에 10분이 경과된 요청은 목록에서 사라지며 응답을 보낼 수 없다.



화면 3. 자리 확인 요청 알림 팝업 모달



화면 4. 자리 확인 응답 대기 목록

- 주요 소스코드

구분	[websocket] 자리 확인 요청 이벤트 처리
코드 위치	<a href="#">reservation-events.gateway.ts</a>
코드	<pre> @SubscribeMessage('user.find-store.server') async userFindStoreToServer(   @ConnectedSocket() socket: Socket,   @MessageBody() userFindStoreServerDTO: userFindStoreServerDTO, ) {   this.websocketLogger.websocketEventLog('user.find-store.server', false, true);   const userId = socket.data.uuid;   const distance = 500;   const {     categories,     numberOfPeople,     delayMinutes,     longitude,     latitude,   }: userFindStoreServerDTO = userFindStoreServerDTO;    // 1. 주점 검색   const stores = await this.storeService.findCandidateStores(     longitude,     latitude,     categories,     distance,   );    // 2. 주점으로 이벤트 전송   for (const store of stores) {     try {       const estimatedTime = await this.dateUtilService.getEstimatedTime(         latitude,         longitude,         store.latitude,         store.longitude,         delayMinutes,       );        const createReservationDTO: CreateReservationDTO = {         numberOfPeople,         estimatedTime,         userId,         storeId: store.id,       };       const reservationId = await this.reservationService.createReservation(createReservationDTO);        const storeSocketId = storeOnlineMap[store.id];       socket.to(storeSocketId).emit('server.find-store.store', reservationId);       this.websocketLogger.websocketEventLog('server.find-store.store', true, true);      } catch (e) {       this.websocketLogger.websocketEventLog('server.find-store.store', true, false);       this.websocketLogger.error(e);       return false;     }   }    return true; } </pre>
설명	<p>① 주점이용자의 현재 위치와의 직선 거리가 500m 이내에 위치한 주점 목록 추출</p> <p>② T-MAP API를 활용하여 주점이용자 현재 위치에서 각 주점까지의 이동 소요 시간을 구해 예상 도착 시간 계산</p> <p>③ Reservation 테이블에 주점별 요청 정보 및 상태 저장</p> <p>④ Online Map에 등록되어 있는 주점업자의 socket id를 통해 client(주점업자)에게 'server.find-store.store' 이벤트 전송</p>



구분	[websocket] 자리 확인 수락 응답 이벤트 처리
코드 위치	reservation-events.gateway.ts
코드	<pre> @SubscribeMessage('store.accept-seat.server') async acceptSeatEvent(   @ConnectedSocket() socket: Socket,   @MessageBody() data: { reservationId: number; userId: string }, ) {   this.websocketLogger.websocketEventLog('store.accept-seat.server', false, true);   try {     const { userId, reservationId } = data;     const reservation = await this.reservationService.getReservationById(reservationId); ①      const { reservationStatus } = reservation;     if (reservationStatus !== ReservationStatus.REQUESTED) {       return {         isSuccess: false,         message: '이미 처리된 요청입니다.',       };     } ②      await this.reservationService.updateReservationStatus(reservationId, 'PENDING'); ③      const userSocketId = userOnlineMap[userId];     socket.to(userSocketId).emit('server.available-seat.user', reservationId);     this.websocketLogger.websocketEventLog('store.accept-seat.server', true, true); ④      return {       isSuccess: true,     };   } catch (e) {     this.websocketLogger.websocketEventLog('store.accept-seat.server', true, false);     this.websocketLogger.error(e);     return {       isSuccess: false,       message: '만료된 요청입니다.',     };   } } </pre>
설명	<p>① Reservation 테이블에서 이벤트 바디로 넘겨받은 reservationId에 해당하는 요청/예약 정보 조회</p> <p>② 조회한 Reservation의 상태가 'REQUESTED(요청)'가 아닌 경우 이미 처리된 요청임을, 데이터가 존재하지 않을 경우 만료된 요청임을 알리는 에러 핸들링</p> <p>③ 요청 수락의 의미로 해당 Reservation 데이터의 reservationStatus(상태) 칼럼 값을 'PENDING(대기)'로 변경</p> <p>④ Online Map에 등록되어 있는 주점이용자의 socket id를 통해 client(주점이용자)에게 'server.accept-seat.user' 이벤트 전송</p>

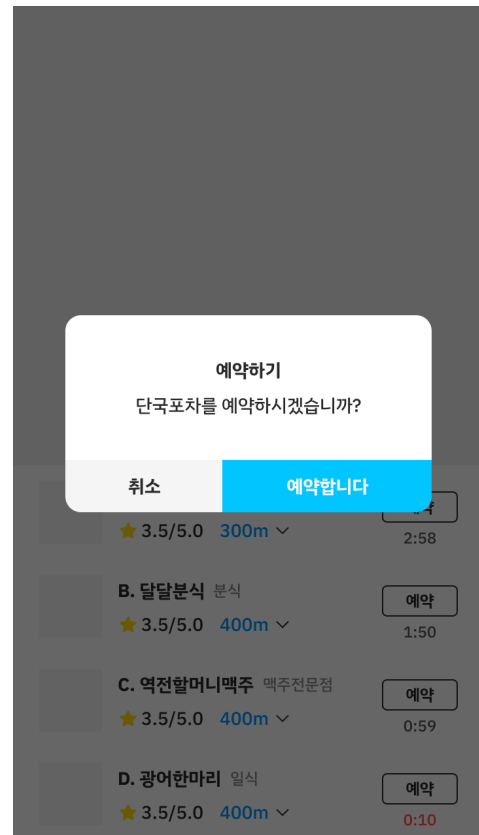
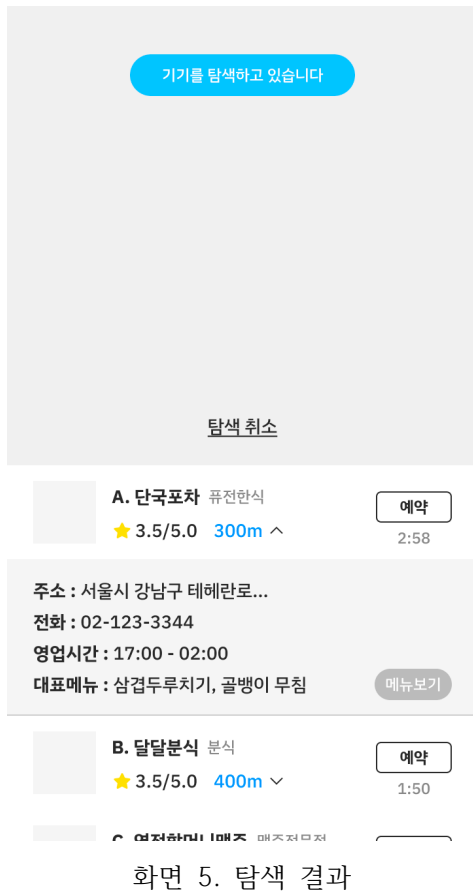
## 2. 실시간 예약 기능

### - 주점이용자

<화면 5> 수락 응답을 보낸 주점에 대하여 주점이용자는 실시간으로 추가되는 해당 주점의 정보와 위치를 확인할 수 있다. 주점이용자는 지도상에 표시되는 주점의 위치, 거리, 메뉴 등을 고려하여 원하는 주점을 선택하여 예약한다. 이때, 주점업자의 응답 시점으로부터 3분 동안만 예약을 요청할 수 있다. 각 응답 주점 목록에 표시되는 3분 카운트다운이 종료되면 해당 주점은 목록에서 사라지

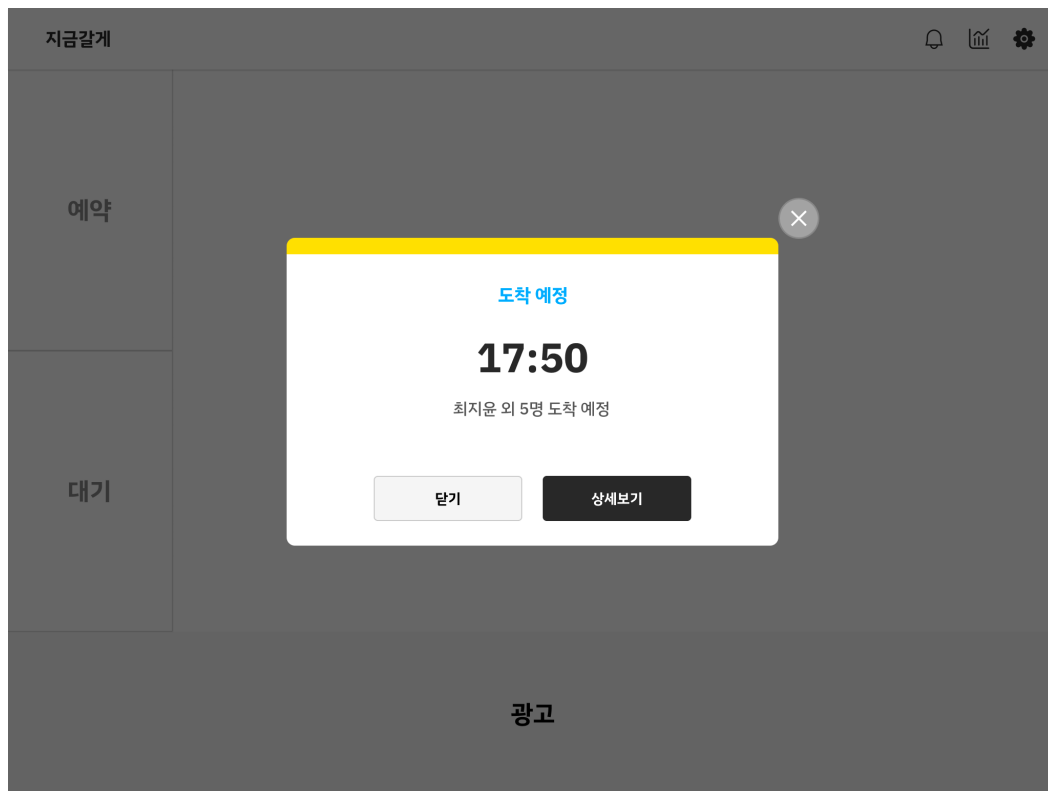
며, 예약할 수 없다.

<화면 6> 선택한 주점을 확인하는 과정을 거친 후 예약이 체결된다.



#### - 주점업자

<화면 7> 예약이 체결되면 주점업자에게 즉시 알림 팝업이 전송된다. 예약자의 도착 예정 시간 정보 등을 확인한다.



화면 7. 예약 체결 알림 팝업 모달

- 주요 소스코드

구분	[websocket] 예약 이벤트 처리
코드 위치	<a href="#">reservation-events.gateway.ts</a>
코드	<pre> @SubscribeMessage('user.make-reservation.server') async makeReservationEvent(   @ConnectedSocket() socket: Socket,   @MessageBody() data: { storeId: string; reservationId: number }, ): Promise&lt;boolean&gt; {   this.websocketLogger.websocketEventLog('user.make-reservation.server', false, true);   const { storeId, reservationId } = data;   try {     await this.reservationService.updateReservationStatus(reservationId, 'RESERVED'); ①      const storeSocketId = storeOnlineMap[storeId];     socket.to(storeSocketId).emit('server.make-reservation.store', reservationId); ②     this.websocketLogger.websocketEventLog('server.make-reservation.store', true, true);      return true;   } catch (e) {     this.websocketLogger.websocketEventLog('server.make-reservation.store', true, false);     this.websocketLogger.error(e);      return false;   } } </pre>
설명	<p>① 주점이용자가 선택한 주점에 대하여 Reservation 테이블에 저장되어 있는 해당 데이터의 reservationStatus(상태) 칼럼 값을 'RESERVED(예약)'로 변경하여 저장</p> <p>② Online Map에 등록되어 있는 주점업자의 socket id를 통해 client(주점업자)에게 'server.make-reservation.store' 이벤트 전송</p>


### 3. 예약 확인 및 관리 기능


#### - 주점이용자

<화면 9> 예약 체결 후 주점이용자의 홈 화면에는 현재 진행 중인 예약 건이 보여진다. 간단한 주점 정보와 예정된 출발 시간 안내를 확인할 수 있다.

<화면 10> '더보기' 버튼을 누르면 상세한 주점 정보 확인 및 예약 관리를 위한 화면이 나타난다. 주점 대표번호로 전화걸기, 길찾기 연결, 도착 예정 시간 변경(연장) 신청, 예약 취소 기능도 사용할 수 있다.

## 지금갈게


 예약완료



단국포차 퓨전한식  
 ★ 3.5/5.0 300m


출발까지 15분 남았습니다.
 [더보기](#)


+ 실시간 예약

 홈
  찜
  이용내역
  My 지금갈게

화면 8. 예약 확인 (홈)

## 지금갈게

 예약완료



단국포차 퓨전한식  
 ★ 3.5/5.0 300m

출발까지 15분 남았습니다.
 [더보기](#)

---

주소 : 서울시 강남구 테헤란로...  
 전화 : 02-123-3344  
 영업시간 : 17:00 - 02:00  
 대표메뉴 : 삼겹두루치기, 골뱅이 무침, 버섯된장 >





전화 걸기

길 찾기

시간 변경

예약 취소

+ 실시간 예약

 홈
  찜
  이용내역
  My 지금갈게

화면 9. 예약 상세 및 관리 (홈)

### - 주점업자

〈화면 8〉 도착 이전 상태인 모든 예약 체결 건들은 ‘예약’ 메뉴에서 확인할 수 있다. 예약자의 현재 위치 확인, 예약 취소, 예약자의 도착을 알리는 체크인 기능도 사용할 수 있다.

예약	<div>오후 5시 45분</div>	<div>최지윤 외 5명 010-1234-5678 신용등급 : 78점</div>	<div>위치확인</div>	<div>취소</div>	<div>체크인</div>
대기					
광고					

화면 10. 체결 예약 목록

- 주요 소스코드

구분	[GET Method] 주점이용자/주점업자 현재 체결된 예약 내역 반환
코드 위치	<a href="#">reservation.controller.ts</a>
코드	<pre> ① @Get('user/:id') async getUserReserved(   @Query('status') status: string,   @Param('id') userId: string, ): Promise&lt;ReservationDTO&gt; {   const reservation = await this.reservationService.getReservationByUserId(status, userId);    return reservation; }  ② @Get('store/:id') async getStoreReservedAndPending(   @Query('status') status: string,   @Param('id') storeId: string, ): Promise&lt;ReservationDTO[]&gt; {   const reservations = await this.reservationService.getStoreReservationByStatus(status, storeId);    return reservations; } </pre>
	① 주점이용자가 선택한 주점에 대하여 Reservation 테이블에 저장되어 있는 해당 데이터의 reservationStatus(상태) 칼럼을 'RESERVED(예약)'로 변경하여 저장
설명	② Online Map에 등록되어 있는 주점업자의 socket id를 통해 client(주점업자)에게 'server.make-reservation.store' 이벤트 전송

## □ 결과물 형태 및 서비스 방식

- 결과물 형태
  - 주점 이용자 측: 스마트폰 애플리케이션
  - 자영업자 측: 태블릿 애플리케이션
- 서비스 방식
  - 서비스 시연 동영상
  - Google Play 스토어 및 앱스토어 업로드
  - 웹 서비스 제공을 위한 전용 웹 사이트

## □ 기대효과 및 활용분야

- 기대효과

<b>사용자 측면</b>	<ul style="list-style-type: none"> <li>- 주점 이용자는 실시간 예약 서비스를 통해 간편하게 근처에 자리가 있는 식당을 한눈에 확인 가능</li> <li>- 간편하게 고객의 예약 요청을 확인 및 응답을 할 수 있는 주점업자는 담당 직원을 둘 필요가 없고, 매장 운영에 있어서 시간적 비용 절약</li> <li>- 노쇼를 서비스적으로 관리해주기 때문에 주점업자의 부담 감소</li> </ul>
<b>비즈니스 측면</b>	<ul style="list-style-type: none"> <li>- 실시간 예약 서비스를 통해 고객 식당 선택의 폭을 넓혀 고객을 넓은 범위의 식당으로 분산</li> <li>- 코로나로 매출 타격을 받은 소상공인에 대하여 상권 활성화를 통한 주점업의 전반적인 경기침체 위기 극복 기대</li> </ul>
<b>개발자 측면</b>	<ul style="list-style-type: none"> <li>- 실시간 대용량 트랜잭션 처리를 위한 시스템 및 프로그래밍 설계 및 개발</li> <li>- Pub/Sub 구조의 이해와 비동기 이벤트 기반 서비스 구조의 이해</li> <li>- 모바일 하이브리드 앱 개발 및 사용자 친화적인 UI 설계</li> <li>- Node.js(NestJS) 기반 확장성 있는 실시간성 아키텍처 설계 및 개발</li> <li>- GPS 기반 현재 사용자 위치에 따른 정보 제공으로 AWS Location 등과 같은 위치 서비스 활용</li> </ul>

#### ○ 결과물 활용 방안

<b>사업화 계획 및 비즈니스 모델 구축</b>	주점 이용자가 보는 화면에 광고 게시와 주점업자 서비스 구독료를 이용한 수익 모델을 구축할 수 있다. 쉽게 원하는 주점에 들어갈 수 있다는 장점을 부각하여 충성고객을 유치할 수 있다. 이렇게 사용자가 많아짐에 따라 주점업자에게 광고를 의뢰받아 메인 화면에 광고를 게시할 수 있다. 이후 주점 외 오락 시설(노래방, 피시방) 등으로 확장하여 타겟 고객층을 넓히고 해당 주점에 서비스 구독료를 받는 방식으로 비즈니스 모델 확장이 가능하다. 또한 실시간 예약 시스템을 구축하여 기존의 중/장기적인 예약만을 다루고 있던 예약 서비스들이 축적하지 못한 예약, 주점 이용자, 주점업자 db를 활용하여 다른 예약 서비스와 차별화된 통합 예약, 매장관리 플랫폼을 구축할 수 있으며, 개인화된 근처 식당 추천 서비스로 사업화가 가능하다.
<b>비즈니스 측면 개발자 측면</b>	머신러닝 기업과의 추천 시스템 기술 제휴가 가능하며, 예약 시 간편한 결제와 할인을 핀테크 기업, 금융 기업과의 제휴를 통해 실현할 수 있다.

#### □ 기획심의 개선 요구사항 반영 결과(주요)



개선 요구사항	반영 결과
타겟 지역을 선정하여 많은 주점을 확보하는 것이 중요하겠다.	팀 모두가 대학생임을 활용하여 1차 target을 대학가 주점으로 선정. 대학가 주점 특성상 가까운 거리에 모여있으며 즉흥적으로 주점에 자리를 찾는 상황 다수 존재. 또한 주요 고객이 대학생이기 때문에 학교 커뮤니티를 활용한 마케팅 용이.
매칭 성공률이 저조할 것 같다.	배달의 민족 사장님 서비스와 유사한 소리가 나는 푸시 알림 제공. 또한 바쁜 시간에도 예약 요청을 쉽게 확인하고 응답할 수 있게 큰 버튼, 큰 글씨, 간단한 조작법의 사용자 친화적인 UI/UX 제공.
예약 방식이 어렵다.	기존의 전화 예약방식은 자리 확인부터 예약까지 평균 60초 내외의 시간 소요. 전화를 받는 상황에서 주점 사장님은 다른 업무 불가능. 이를 자리 있다는 버튼 한번 누르는 행동으로 전환하며 예약 방식을 개선. 확인부터 예약까지 2초 내외로 소요.
어떻게 참여 유도를 할지 고민해야한다. 비즈니스 모델을 구체적으로 구축해야한다.	초기에는 최소한의 수익 모델을 가지고 사용자 유치에 집중할 계획. 광고 수수료와 낮은 구독료, 고객의 지각에서 나오는 수수료를 가게에 전달함으로써 가게 확보. 이후 참여자가 많아지고 매칭 성공률이 높아지면 세부적인 구독료(프리미엄, 일반), 고객-가게 중개 수수료, 광고 수수료로 비즈니스 모델 구축.