

# 가게 고객 대기 관리를 위한 서비스 개발 프로젝트

A service development project for store waiting management

문승훈(Seunghun Moon), 조예린(Yerin Cho),  
손영호(Youngho Son)

## 요약

가게 입구에 서서 대기하는 모습은 인기 있는 매장에서 흔히 볼 수 있는 광경이다. 이 때, 대기 공간이나 시스템이 구축되어 있지 않아 많은 사람들이 불편을 겪고 있다. Hate-Wait 서비스는 이러한 문제를 해결하여 가게와 손님 모두에게 편의성을 제공하기 위해 기획되었다. Android Application, Web, Server를 모두 새로 구축하였다. 본 서비스를 통해 가게 업주는 손쉽게 고객 대기 관리 시스템을 구축할 수 있으며, 손님은 시공간 제약으로부터 자유로워진다. 본 논문에서는 Hate-Wait 서비스의 개발 과정, 결과, 사용방법과 한계점 등을 명시한다.

**키워드:** 고객 대기 구축, 대기열, 리액트, 서버, 안드로이드 앱

## Abstract

There are cases where a store number is required. At this time, many people are inconvenienced by the lack of a standby space or a system. The Hate-Wait service was designed to solve these problems and provide convenience to stores and customers. Android App, Web, and Server are all newly built. By this service, the owner of the shop can easily construct the customer stand-by management system, and the customer is free from the restriction of space and time constraints. This paper clarifies the development process, results, usage and limitations of the Hate-Wait service.

**Keyword:** Android app, Build customer queue, react, server, queue

## 1. 서론

21세기 4차 산업혁명과 더불어 우리는 통신의 황금기를 맞았고, 그 중에서도 5G 기술은 최대 1,000Mbps(즉, 1Gbps)의 데이터 전송 속도, 대기 시간 감소, 원격 지원 서비스 범위 확대 이외에도 상상할 수 없을 만큼 많은 이점을 제공한다[1]. 반면, 가게, 전사회와 같은 오프라인 공간에서 아직

‘대기’라는 과정에 통신 서비스를 제공하고 있지 않은 부분들이 다수 존재하며, 위 같은 상황은 기술의 부족으로 생겨난 것이 아님을 알 수 있다. 또한, 위 같은 과정에서 주변 상권 혹은 주민들로 하여금 ‘민원’과 같은 새로운 문제를 낳기도 하며, 이로 인해 서비스 제공 자체에 문제가 생기는 사례[2]도 존재한다. 이러한 사례들이 존재함에도 서비스 제공자들은 대기의 공평함과 서비스 지속을 위해 추가적인 업무를 진행하지 않으며, 서비스를 이용하고자 하는 고객은 이러한 불편을 지속해야 하는 상황이다.

본 팀은 위에서 언급한 문제를 해결하기 위해 공간과 시간의 제약을 줄이기 위한 대기 관리 서비스 시스템 Hate-Wait을 개발하였다. 위 서비스는 전문적인 지식이 없는 일반인들도 사용할 수 있도록, 안드로이드 앱과 웹 클라이언트로 서비스를 제공한다. 본 논문에서는 Hate-Wait 서비스 개발과정과 내용, 사용 방법, 각 클라이언트 별 특성과 더불어 결론 및 향후 연구에 대하여 설명한다.

## 2. 관련 연구

### 2.1 대기행렬이론

대기행렬이론은 대기행렬을 수학적으로 다루는 이론으로, 대기행렬에 도착하는 것, 대기하는 것, 그리고 서비스 과정의 일련의 프로세스들에 대한 수학적, 확률적 분석이 가능케 한다.

대기행렬이론은 과거부터 많은 연구 사례가 존재하며, 시스템 별 특성을 통해 산출모형을 연구한 대기행렬 예측정보 산출모형에 관한 연구(이청원, 2001) [3]부터 주유소와 관련한 셀프서비스의 최적 운영모형 개발(조재형, 2018) [4] 까지 다양한 연구가 이뤄진 가운데, 우리는 시스템특성에 영향을 미치는 요소에 공간의 크기와 고객의 방문 시간이 중요한 요소임을 알 수 있다.

고객의 수	$L_q = \frac{\lambda^2}{\mu(\mu - \lambda)}$
대기행렬에서 평균적으로 기다리는 시간	$W_q = \frac{\lambda}{\mu(\mu - \lambda)} = \frac{L_q}{\lambda}$
평균적으로 기다리는 시간	$W = \frac{L}{\lambda} = W_q + \frac{1}{\mu}$
시스템 내에서 고객의 평균 수	$L = L_q + \frac{\lambda}{\mu}$
n명의 고객이 있을 확률	$p_n = \left(1 - \frac{\lambda}{\mu}\right) \left(\frac{\lambda}{\mu}\right)^n$
서비스 이용률 (Utilization, $\rho$ )	$\frac{\lambda}{\mu}$

그림 1 대기행렬 이론에서의 수학적 모델

이 때, 가게마다 고객이 존재하는 평균 수는 다를 수 있으나, Hate-Wait 서비스는 공간에 대한 제약을 가게의 공간으로 한정하는 것이 아닌 대기할

수 있는 모든 공간으로 확장하는 방법을 통해 시스템 내에 대기할 수 있는 고객의 범위를 넓혔으며, 이는 대기과정에서 일어날 수 있는 비용의 축소, 시스템을 지속적으로 운용할 수 있는 효율성 측면에서 효과적이다.

### 3. 사용 방법

3장 사용 방법에서는 Android Application 및 React based Web의 Hate-Wait의 실제 사용 방법에 대해 기술하며, 각 액티비티와 페이지의 상세한 설명을 위해 실제 서비스 과정을 담은 그림을 함께 첨부하였다.

#### 3.1 Android App

Android App을 통해 서비스에 접근하는 고객은 가게 회원과 손님 회원으로 구분되며 회원가입 또는 로그인 시 계정 유형을 선택하여 각각의 메뉴로 진입할 수 있다.

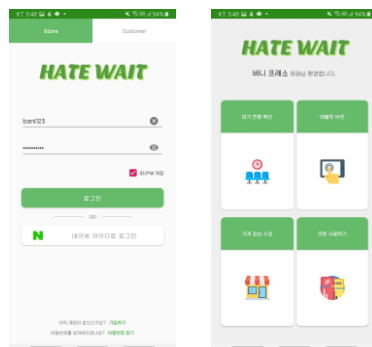


그림 2 로그인 화면 (좌), 가게 메인 메뉴 (우)

가게 회원은 대기 등록 메뉴를 통해 가게 앞에 비치된 태블릿 기기로부터 방문 손님들의 대기 등록을 받을 수 있다. Hate-Wait 서비스에 가입된 회원이 대기 등록을 하는 경우 아이디와 인원 수를 입력 받으며 가입된 회원 이름의 일부를 통해 사용자로부터 아이디 확인 후 대기 등록이 완료된다. 비회원의 경우 이름, 전화번호, 인원 수를 입력하면 대기 등록이 완료된다.

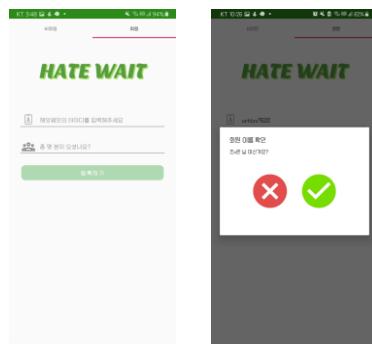


그림 3 대기 등록 화면(좌), 회원 아이디 확인 다이얼로

그 (우)

대기 현황 확인 메뉴를 통해 현재 가게에 대기 중인 손님 리스트를 확인할 수 있다. 목록을 눌러 해당 손님의 최근 호출 시각을 확인할 수 있으며, 좌측의 중 모양 버튼을 눌러 SMS 혹은 상태바 Push 알림을 통한 손님 호출이 가능하다. 또한, 목록을 왼쪽으로 슬라이드 하여 해당 손님을 대기 목록에서 삭제할 수 있다. 오른쪽 하단의 + 버튼을 통해 본 페이지에서도 대기 손님 등록이 가능하며 이는 비회원 대기 등록과 동일한 동작을 한다.

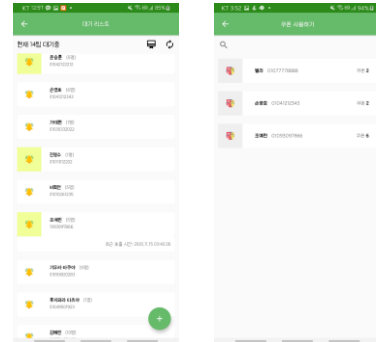


그림 4 대기 현황 화면 (좌), 쿠폰 사용 화면 (우)

쿠폰 사용 페이지는 가게 메인 메뉴 또는 대기 리스트 페이지의 오른쪽 상단 버튼을 통해 진입할 수 있다. 해당 가게에 쿠폰을 보유하고 있는 전체 손님 목록이 최근 방문순으로 보이며 상단 검색바를 통한 회원 이름 검색이 가능하다.

손님 회원의 경우 메인 화면에서 본인의 대기 현황을 확인하거나 진행중인 대기를 취소할 수 있다. 또한, 쿠폰 현황 확인하기 메뉴를 통해 스탬프와 쿠폰을 보유하고 있는 가게 목록을 확인할 수 있으며 해당 가게의 보유 스탬프, 쿠폰 혜택, 유효기간 등 세부적인 정보를 확인할 수 있다.

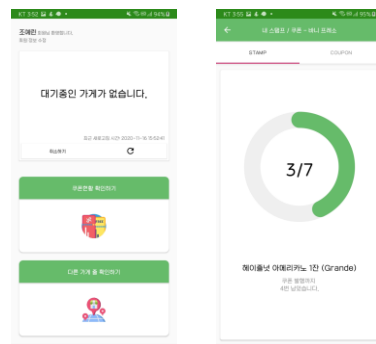


그림 5 손님 메인 메뉴 (좌), 스탬프 보유 현황 (우)

다른 가게 줄 확인하기 메뉴에 최초 진입 시 사용자에게 위치 정보 권한을 요청한다. 사용자의 현위치를 중심으로 Hate-Wait에 등록된 회원 가게와 비회원 가게가 지도에 마커로 구분되어 표시되며 사용자가 원하는 위치로 지도 영역을 이동시킬 때마다 정보가 업데이트된다. 마커 클릭시 영업 시

간, 현재 대기 인원 등 가게에 대한 세부 정보를 확인할 수 있다. 사용자가 위치 권한 거부 시 지도를 현위치로 이동하는 기능은 제한된다.

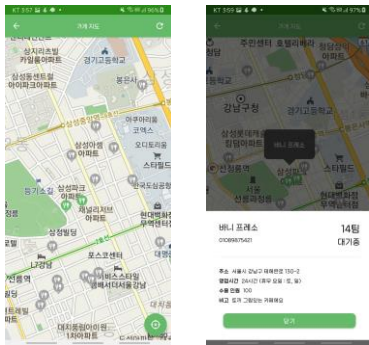


그림 6 다른 가게 줄 확인 (좌), 지도상 가게 정보 (우)

가게 회원과 손님 회원은 이메일, 아이디, 이름을 제외한 모든 회원 정보를 수정할 수 있다. 특히, 가게 회원의 경우 회원 정보 수정 액티비티를 통해 쿠폰 서비스의 사용 여부와 스탬프 개수, 혜택, 유효 기간 등을 설정할 수 있다.

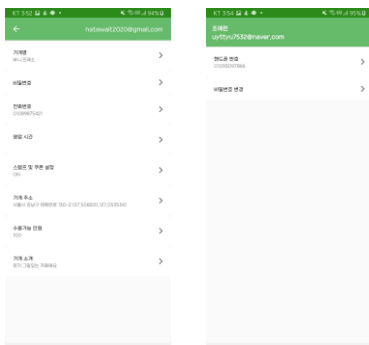


그림 7 가게 정보 수정 (좌), 회원 정보 수정 (우)

### 3.2 React Web

Web 서비스 대상은 가게 업주이며 Chrome, Safari, Internet Explorer 등 다양한 웹 브라우저를 지원한다.

본 서비스 웹 사이트의 첫 페이지는 로그인 페이지이다. 본 페이지에서 회원정보가 저장된 사용자는 로그인 버튼을 통해 서비스에 접근할 수 있으며, 회원정보가 없는 사용자는 로고 아래의 카테고리를 통해 웹 기능의 설명을 확인할 수 있다. 회원가입 버튼을 통해 회원가입 페이지로 이동할 수 있다.



그림 8 로그인 전 메인 화면 (좌), 회원가입 페이지 (우)

회원가입 과정에서 사용자는 서비스를 이용할 때 사용할 아이디와 비밀번호에 대한 중복 검사를 진행하고, 이메일과 전화번호 등의 정보를 입력하는 과정을 필수로 거친다. 이후, 가게 주소, 운영시간을 입력하여 손님 사용자에게 제공할 정보를 저장한 후, 모든 과정에서 올바른 양식과 서버 데이터베이스에 존재하는 회원과 중복되지 않는 정보를 입력한 사용자만이 성공적으로 회원가입을 완료한다.

회원가입에서 입력한 정보를 통해 올바르게 로그인한 경우, 저장된 가게 정보를 홈페이지에서 확인할 수 있으며, 고객 통계, 대기 현황 확인, 쿠폰 서비스 등의 기능이 활성화된다.



그림 9 로그인 페이지(좌), 메인 페이지(우)

권한이 활성화되어 로고 아래의 카테고리로 기능에 접근하는 경우, 가게의 대기 현황, 쿠폰을 가진 사용자, 고객의 방문 통계 데이터를 확인할 수 있으며, 대기 현황에서는 표 우측의 중 모양 버튼을 통해 고객을 호출할 수 있다. 호출 시, 시간이 표에 표시되며, Hate-Wait 서비스는 대기 고객이 입력한 전화번호로 SMS 메시지를 송신한다.



그림 10 가게 고객 대기 현황 페이지

카테고리의 CUSTOMER 버튼을 누르면, 사용자는 고객 방문 통계 페이지로 이동할 수 있다. 본 페이지에서는 가게 방문 고객 추이를 그래프 모듈을 통해 시각적으로 확인할 수 있으며, 각 그래프 바에 마우스오버시, 세부적인 인원 수를 확인할 수 있다.



그림 11 고객 통계 페이지

카테고리의 COUPON 버튼을 누르면, 사용자는 쿠폰 관리 페이지로 이동할 수 있다. 본 페이지에서 쿠폰 보유 회원의 이름과 보유 쿠폰 개수 목록을 리스트 형태로 확인할 수 있다. 표의 우측 쿠폰 아이콘을 클릭하여 사용자 쿠폰을 사용할 수 있으며, 이 때, 가게마다 정해 놓은 혜택을 고객에게 제공해야 한다.

이름	전화번호	쿠폰 번호	보유 개수
김민준	010-2017888	2020-10-18T08:57:49.000Z	6
박민준	010-7778888	2020-10-18T08:43:44.000Z	2
최민준	010-4121243	2020-10-18T08:56:49.000Z	2

그림 12 쿠폰 페이지

## 4. 설계

본 장에서는 Android App과 React Web 제작을 진행하며, 구축된 전체적인 시스템 구조와 설계 방식과 같은 지식을 기술한다. Hate-Wait의 전체적인 구조는 그림 5와 같으며, 크게 클라이언트와 프록시, 데이터베이스, 서버 모듈로 나뉜다.

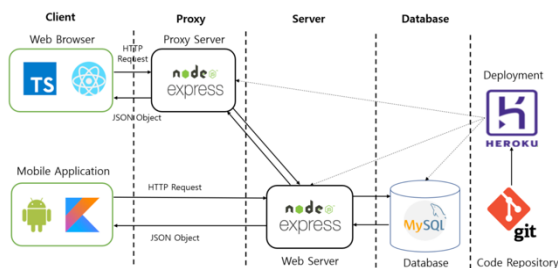


그림 13 Hate-Wait 시스템 구조

## 4.1 클라이언트 (앱, 웹)

본 프로젝트에서 서비스를 구축하기 위해 Android와 React를 사용하였다.

Android는 Android Studio 4.0에 Kotlin으로 개발하였으며, 최소 지원 버전은 Android 5 (API 수준 21), 대상 지원 버전은 Android 9 (API 수준 28)로 설정하였다. 서버에 데이터를 요청하고 응답 받기 위해 REST API 통신 라이브러리인 Retrofit2를 사용하였다. Singleton 패턴을 적용하여 Retrofit2 객체를 한 번만 생성하였으며, data class를 이용해 데이터 전송 개체인 Data Transfer Object(DTO)를 만들어 사용하였다.

React 라이브러리를 사용하여 웹을 구축하기 위해 Visual Studio 개발 도구를 채택하였으며, 추가적으로 npm 모듈과 함께 TypeScript 언어를 사용하였다. 위 과정에서, React의 장점인 재사용을 극대화하기 위해, Atomic Structure를 구성하여 기본 재료가 되는 Atom을 가장 먼저 제작하였고, 이후, 모든 컴포넌트의 상태를 통합적으로 운용하기 위해 React-Redux 모듈을 사용하였다.

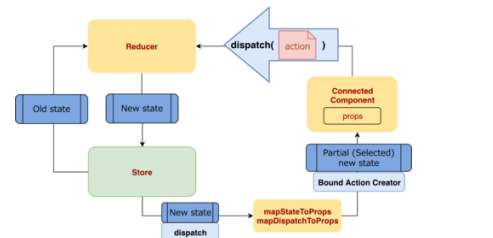


그림 14 Atomic Structure (위) React-Redux 흐름도(아래)

## 4.2 데이터베이스

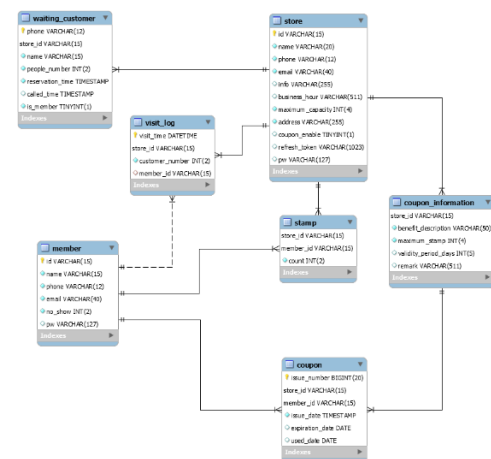


그림 15 Entity-Relationship Diagram

모든 데이터베이스(이하 DB) 쿼리 요청 시 connection pool을 적용하여 DB 서버의 연결 및 해제 시 자원 낭비를 방지하고 연결 수립 대기시간 단축을 도모했다.

DB ERD는 그림9와 같다. 총 7개의 테이블로 구성되어 있다. Member 테이블은 손님 회원, waiting\_customer 테이블은 대기중인 손님으로 회원, 비회원 손님을 is\_member 필드로 구분한다. store는 가게로 기본 가게 정보에 추가적으로 쿠폰 서비스 지원 여부에 따라 하위 자식 테이블인 coupon\_information, coupon, stamp 테이블을 가질 수 있다. 부모 - 자식 관계를 갖는 테이블은 모두 생명주기를 같이하도록 구현했기 때문에 가게 회원이 탈퇴하거나 쿠폰 서비스를 비활성화 할 경우 모든 자식 테이블이 삭제된다. 방문 통계 기능에 사용되는 visit\_log 테이블은 방문 시간, 방문 일행수를 비롯한 가게, 손님 회원의 아이디를 저장한다.

#### 4.3 서버

본 서비스의 웹 서버는 Node.js 를 통해 구현했다. Node.js는 이벤트 지향, 비동기적 프로그래밍을 지원하는 네트워크 서비스 디자인에 특화된 플랫폼으로 시스템 자원을 최대한 사용할 수 있다는 이점이 있다.

실시간 대기열 업데이트 기능에 한정하여 HTTP 비연결성과 무상태성의 한계를 극복하기 위해 Server-Sent Events (SSE)를 사용한다. 클라이언트의 요청 없이 동적으로 웹 페이지 상태를 갱신하는 여러 기술 중 웹소켓과 SSE가 서버의 성능, 하드웨어 비용 면에서 효과적이다.[6] SSE는 웹소켓과 달리 HTTP를 그대로 사용하면서 단방향 연결을 지원하기 때문에 보다 적은 오버헤드로 서버에서 클라이언트로의 데이터 송신이 가능하다. 구현 방식은 그림 16과 같다.

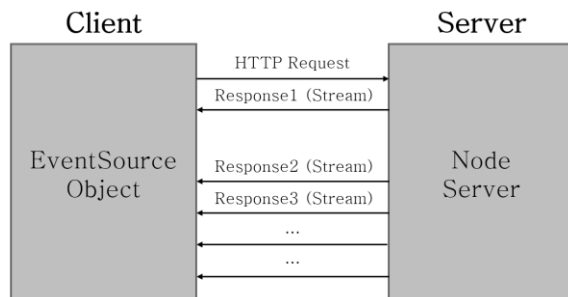


그림 16 SSE Diagram

#### 4.4 프로토콜

본 서비스는 RESTful API 기반 JavaScript Object Notation(JSON)을 사용하여 HTTP 프로토콜을 통해 개발했다.

JSON은 JavaScript에 의해 클라이언트가 추가적

인 작업 없이 파싱 할 수 있어 기존 XML 통신에 비해 더 빠르고 간단하며 UI단의 처리 속도 향상으로 이어진다.

RESTful API는 HTTP 및 Uniform Resource Identifier(URI)을 명세하여 클라이언트가 Create, Read, Update, Delete에 해당하는 HTTP 메소드 POST, GET, PUT, DELETE를 통해 자원에 접근하도록 하는 방식이다.[5]

즉, JSON과 RESTful API 프로토콜 설계로 Android, React 등 서로 다른 플랫폼을 독립적으로 개발할 수 있고 공통의 데이터 파싱 과정을 가지게 됨으로써 모듈의 작업량이 줄어들고 확장성을 갖출 수 있다. 예시 프로토콜 데이터는 표1과 같다.

표1 프로토콜 데이터

대기열 등록	
Method	POST
URI	https://hatewait-server.herokuapp.com/stores/<store-id>/waiting-customers
Request Body (JSON)	{ "member_id" : "<id>", "people_number": "<number>", "is_member" : "<true>" }
Response Body (JSON)	{ "name": "<name>", "count": "<count>" }

#### 5. 결론 및 향후 연구

21세기 통신 기술의 발전은 현대인에게 수많은 편의를 가져다 주었으나, 아직 많은 부분에서 이러한 기술의 장점을 사용하지 않고 있다. Hate-Wait을 통해 대기 관리 시 시공간의 제약을 최소화하여 손님과 가게 모두의 편의성을 확보할 수 있게 된다. 특히, 방문 통계 기능과 쿠폰 서비스를 통한 혜택 등 직접적으로 이를 체감할 수 있다. 하지만, 코로나19 범유행 사태로 사용자 테스트를 거치지 못해 정식 서비스화는 아직 어려운 단계이다. 테스트를 통한 문제점 탐색, 지속적인 기능 개선이 이뤄진다면 경쟁력 있는 서비스로 사업화도 가능할 것이다.

#### 참고 문헌

- [1] Red Hat - 통신 中 5G 네트워크란 - 기술, 속도, 차이점 및 비교
- [2] 이정현, "이웃주민 원성에... '골목식당' 돈가스 집, 결국 포방터 떠나" <연합뉴스>

2019.11.07

- [3] 이청원, 심소정. (2001). 대기행렬 예측정보 산출모형에 관한 연구. 서울시연구, 2(2), 61-77. Yun, S., Han, D., Chun, S., Oh, S. J., Yoo, Y. and Choe, J. (2019) 'CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features', 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE. doi: 10.1109/iccv.2019.00612.
- [4] 조재형, 강태영. (2018). 대기행렬이론을 이용한 기술기반 셀프서비스의 최적운영모델 개발. 인터넷전자상거래연구, 18(3), 265-282. Systems and Technologies (CISTI). IEEE. doi: 10.23919/cisti.2018.8399283.
- [5] Z. Niu, C. Yang, Y. Zhang. "A design of cross-terminal web system based on JSON and REST." 2014 IEEE 5th International Conference on Software Engineering and Service Science, Beijing, 2014
- [6] A. Rasmus, O. Oliver. "Performance comparison of XHR polling, Long polling, Server sent events and Websockets." Faculty of Computing Blekinge Institute of Technology SE-371 79 Karlskrona Sweden, 2017