소프트웨어 요구 분석서

(Software Requirement Specification)

[프로젝트 명 : 배배배(배보다 배꼽이 큰 배달료)]

[팀명 : 슈퍼노바(supernova)]

[201644038, 간명해(PM)

201544102, 최태준(서기)

201644056, 이교범(브레인)

201644086, 우도균(브레인)

201744053, 양한준(중재자)]

- 목 차 -

[1 개 요 3](#_Toc449968624)

[1.1 시스템의 목적 3](#_Toc449968625)

[1.2 범위 3](#_Toc449968626)

[1.3 정의, 약어 4](#_Toc449968627)

[1.4 참조 4](#_Toc449968628)

[2 기능적 요구 6](#_Toc449968629)

[2.1 기능적 요구사항 목록 6](#_Toc449968630)

[2.2 기능적 요구사항 정의 7](#_Toc449968634)

[3 비기능적 요구 6](#_Toc449968629)

[3.1 비기능적 요구사항 목록 6](#_Toc449968630)

[3.2 비기능적 요구사항 정의 7](#_Toc449968634)

[4 기타 요구 및 제약 사항 21](#_Toc449968638)

[4.1 성능 요구 (반응 시간, 처리 소요 시간, 처리율) 21](#_Toc449968639)

[4.2 H/W 요구 (기억 장치 규모, 통신수용도) 21](#_Toc449968640)

[4.3 예외 조건 및 이의 처리 21](#_Toc449968641)

[4.4 자원, 인력에 대한 제약 조건 21](#_Toc449968642)

[5 인수 조건 22](#_Toc449968643)

[5.1 기능 시험 및 성능 시험 22](#_Toc449968644)

[6 참고 자료 22](#_Toc449968645)

# 개 요

## 시스템의 목적

평소 사업에 대한 아이디어를 찾는 도중 배달 이용료(배달팁) 증가에 대한 부정적 기사를 접하였고 자체적으로 지역 커뮤니티에서 배달 팁 인상 또는 이용료 문제에 대해 조사하던 중 배달 비용을 1/N로 나눠서 이용하거나 직접 포장하러 가는 후기를 적지 않게 목격하면서, 1km 미만 거리 내에 있는 사용자들끼리 결합하여 배달비용을 아낄 수 있도록 만든 거리 기반 시스템을 이용한 소셜 커뮤니티와 배달 플랫폼을 결합한 어플리케이션을 기획/개발하게 되었다.

목표시장의 규모 및 상황 : 배달 앱들의 매출액이 해마다 비약적인 성장을 하고있다. 통계청에 따르면 지난 해 모바일 음식 서비스 거래액은 24조 9882억원으로 1년전보다 52.1% 증가하였으며, 2017년 거래액 2조 7326억원과 비교하면 5년 만에 10배 가까이 폭증 하였고 5000만 국민이 월 평균 5회 이상 주문하며 월 배달 주문이 3억 건으로 통계 되었다.

경쟁사 : ‘당근마켓’은 중고거래 앱에서 지역 커뮤니티로 발전을 도모하고 있다. 하지만 중고거래 라는 아이덴티티와 이용자의 인식을 없앨 수 없고 지역 관련 포괄적인 글들이 올라오기 때문에 ‘배달료’라는 타겟을 놓고 봤을 때 이 부분에 집중 할 수 없다. 또한 경기도에서 시행하고 있는 ‘공공배달앱’은 매 년 적자에 배달비를 세금으로 충당하기 때문에 현실적이지 않은 방법이다.

목표시장의 특성 : 배달비가 싸다고 해서 사람들이 아무 음식점에서 주문하기 보다는 기존에 자신이 먹어본 맛집을 이용한다. 소비자들이 맛집에서 1시간 넘게 줄을 서는 것도 결국은 그 비용을 다 감수하고 라도 맛집에서 먹으려는 의지가 있다고 생각한다.

향후 전망 및 성장성 : 배달 수요가 계속 증가하는 한 향후 전망은 긍정적이다. 목표 시장은 더욱 커질 것이고 배달비에 대한 불만이 점점 고조 되며 소비 자들이 우리 어플과 같은 다른 대책을 찾을 것이다. 우리 어플이 출시 되고 앱 이용자가 많아지면 지역 커뮤니티를 형성하여 지역에 대한 다양한 이야기, 소개, 광고 등 다양한 컨텐츠를 담을 계획이다.

고객 특성 : 남녀노소 모두 가능하며, 특히 배달을 많이 시켜 먹는 사회 초년생 1인가구와 생활비에 직접 연관이 큰 주부층에서 많이 이용할 것으로 판단 된다.

## 범위

PC는 Window10에서, 모바일은 Android, iOS에서 동작하는 소프트웨어를 개발한다. 개발 환경은 VSCODE를 사용하고, 개발 언어는 Python을 사용한다. 데이터를 저장하고 활용하기 위해 AWS와 Mysql을 이용하며 이외의 소프트웨어를 추가해서 원활한 개발환경으로 진행한다.

## 정의, 약어

|  |  |
| --- | --- |
| 용어 | 설명 |
| SE | (소프트웨어 공학; Software Engineering) 소프트웨어의 개발, 보완, 신뢰성 등을 공학적인 방법으로 연구하는 학문. |
| Android / iOS | 리눅스 커널 기반으로 Google에서 제작하고 있는 스마트폰과 같은 플랫폼의 모바일 운영체제와 미들웨어 및 중요 애플리케이션이 포함된 소프트웨어의 집합. / Apple이 생산하는 제품 중 모바일 기기에 탑재되는 독자 운영 체제. |
| Mysql | 1995년에 발표 된 오픈 소스 DBMS. |
| Flutter(Dart) | 구글에서 2017년 5월 출시된 모바일/웹/데스크톱 크로스 플랫폼 GUI SDK |
| Swift | 애플의 iOS와 macOS를 위한 프로그래밍 언어. |
| Kotlin | IntelliJ IDEA의 개발사 JetBrains에서 2011년에 공개한 오픈 소스 프로그래밍 언어. |
| Swagger | 개발자가 REST 웹 서비스를 설계, 빌드, 문서화, 소비하는 일을 도와주는 대형 도구 생태계의 지원을 받는 오픈 소스 소프트웨어 프레임워크. |
| Python | 1991년에 발표된 인터프리터 방식의 프로그래밍 언어. |
| AWS | 다른 웹 사이트나 클라이언트측 응용 프로그램에 대해 온라인 서비스를 제공. |

## 참조

Android: <https://ko.wikipedia.org/wiki/%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9C_(%EC%9A%B4%EC%98%81_%EC%B2%B4%EC%A0%9C)>

Mysql : <https://ko.wikipedia.org/wiki/MySQL>

Flutter : <https://ko.wikipedia.org/wiki/%ED%94%8C%EB%9F%AC%ED%84%B0>

Swift : https://ko.wikipedia.org/wiki/%EC%8A%A4%EC%9C%84%ED%94%84%ED%8A%B8\_(%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D\_%EC%96%B8%EC%96%B4)

Kotlin : https://ko.wikipedia.org/wiki/%EC%BD%94%ED%8B%80%EB%A6%B0\_(%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D\_%EC%96%B8%EC%96%B4)

Swagger : https://ko.wikipedia.org/wiki/%EC%8A%A4%EC%9B%A8%EA%B1%B0\_(%EC%86%8C%ED%94%84%ED%8A%B8%EC%9B%A8%EC%96%B4)

Python : https://ko.wikipedia.org/wiki/%ED%8C%8C%EC%9D%B4%EC%8D%AC

AWS : https://ko.wikipedia.org/wiki/%EC%95%84%EB%A7%88%EC%A1%B4\_%EC%9B%B9\_%EC%84%9C%EB%B9%84%EC%8A%A4

# 기능적 요구

## 기능적 요구사항 목록

| 번호 | 시스템명 | 업무구분 | 단위기능 | 비고 |
| --- | --- | --- | --- | --- |
| REQ01 | 시스템 성능개선 | 제공 | 회원가입 및 로그인 기능 | 개선 |
| REQ02 | 앱 서비스  시스템 | 제공 | 범위 설정 | 신규 |
| REQ03 | 제공 | 안전 페이 | 신규 |
| REQ04 | 제공 | 유저 평가 | 신규 |
| REQ05 | 제공 | 공동 주문모집 기능 | 신규 |
| REQ06 | 제공 | 올인원 공동 주문 모집 기능 | 신규 |
| REQ07 | DB관리 시스템 | 입력 | 개인정보 관리 기능 | 신규 |
| REQ08 | 제공 | 음식점 정보 제공 | 신규 |

## 기능적 요구사항 정의

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 요구사항 ID | | 요구사항 명 | | 구분 | | 시스템명 |
| REQ01 | | 회원가입 및 로그인 기능 | | 개선 | | 시스템 성능개선 |
| 요  구  사  항  내  역 | 설명 | - 회원가입 및 로그인 기능 개선  • 현재 : ID, Password와 더불어 개인정보를 사이트에 직접 입력하는 방식  • 개선요구 : SNS 및 Portal 계정을 이용한 가입 및 로그인 기능 추가 | | | | |
| 해결안 | - 카카오톡, 네이버 계정을 이용한 회원가입 및 로그인 기능 추가  - 카카오톡: 카카오 개발자 센터(<https://developers.kakao.com>)에서 API 키를 받아 회원가입 및 로그인 기능 구현  - 네이버: 네이버 개발자 센터(<https://developers.naver.com/main/>)에서 API 신청 후 회원가입 및 로그인 기능 구현 | | | | |
| 승인  기준 | 배꼽 회사 | | | | |
| 제약  사항 |  | | | | |
| 변경  내역 | - 요구사항 추가로 인한 변경 | | | | |
| 관련  부서 | 개발 1팀 | 출처 | |  | |
| 테스트  가능성 | 가능 | 난이도 | | 상 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 요구사항 ID | | 요구사항 명 | | 구분 | | 시스템명 |
| REQ02 | | 범위설정 | | 신규 | | 앱 서비스 시스템 |
| 요  구  사  항  내  역 | 설명 | * 사용자 위치 조회 및 위치 변경 기능 * 현재 위치 조회하여 , 게시글 작성시 현재 위치가 포함되어짐 * 사용자가 커뮤니티 게시판 클릭시, 현재 위치 조회하여 근방 1km내외 등록된 게시글을 표시함. | | | | |
| 해결안 | -네이버 지도 API를 이용하여 원하는 거리 설정 가능 (200,500,1000)  -현재 위치가 정확하지 않다면 사용자가 지도에서 본인 위치 선택가능 | | | | |
| 승인  기준 | 배꼽 회사 | | | | |
| 제약  사항 |  | | | | |
| 변경  내역 | - 요구사항 추가로 인한 변경 | | | | |
| 관련  부서 | 개발 1팀 | 출처 | |  | |
| 테스트  가능성 | 가능 | 난이도 | | 상 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 요구사항 ID | | 요구사항 명 | | 구분 | | 시스템명 |
| REQ03 | | 안전페이 기능 | | 개선 | | 시스템 성능개선 |
| 요  구  사  항  내  역 | 설명 | - 안전페이 관리 감독  • 현재 : 유저끼리 결제금액 결제자 계좌에 송금하여 결제 진행하는 상황.  • 개선요구 : 유저 별 결제금액을 회사에서 송금 받은 후 음식 도착 시, 주문  유저들이 도착확인 버튼 누를 시, 결제 금액 결제자에게 송금. | | | | |
| 해결안 | - 유저 결제 금액을 받을 은행 선택 파트너쉽 맺기  - 각 유저 결제 금액 및 실시간 송금 확인을 위한 DB 연동 및 관리  - 커뮤니티 글 작성시 어떤 결제로 진행되는지 선택 가능. 결제 종류에 따라서 글제목에 결제 방법 표시. | | | | |
| 승인  기준 | 배꼽 회사 | | | | |
| 제약  사항 |  | | | | |
| 변경  내역 | - 요구사항 추가로 인한 변경 | | | | |
| 관련  부서 | 개발 1팀 | 출처 | |  | |
| 테스트  가능성 | 가능 | 난이도 | | 상 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 요구사항 ID | | 요구사항 명 | | 구분 | | 시스템명 |
| REQ04 | | 유저 평가 기능 | | 개선 | | 시스템 성능개선 |
| 요  구  사  항  내  역 | 설명 | - 유저 평가 기능  • 현재 : 명확한 유저 평가 기능이 없음  • 개선요구 : 주문 완료 후 ,각 주문에 대한 유저에 대한 평가를 주문 참여 유저들끼리 점수로 0~100점까지 줄 수 있고, 초기에는 50점으로 시작. | | | | |
| 해결안 | - 앱 내에서 유저 클릭 시 해당 유저 평균 점수 확인 가능하도록 DB에 데이터 유저 평가 데이터 추가  - 모집 글 선택시 모집자 점수 확인 가능 및 참여자들 평균 점수 확인 가능하도록  인터페이스 구현 | | | | |
| 승인  기준 | 배꼽 회사 | | | | |
| 제약  사항 |  | | | | |
| 변경  내역 | - 요구사항 추가로 인한 변경 | | | | |
| 관련  부서 | 개발 1팀 | 출처 | |  | |
| 테스트  가능성 | 가능 | 난이도 | | 상 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 요구사항 ID | | 요구사항 명 | | 구분 | | 시스템명 |
| REQ05 | | 공동 주문 모집기능 | | 신규 | | 앱 서비스 시스템 |
| 요  구  사  항  내  역 | 설명 | * 주체자가 주문 게시글 생성 가능 * 제목, 인원수, 위치, 음식 카테고리를 확인 할 수 있는 기능 제공 * 참여자는 현재 위치 근방 500m~1km 내외에 등록된 게시글 조회 가능 * 작성된 게시글 선택시 해당 게시글 오픈 톡방으로 참여자들 끼리 소통 가능. | | | | |
| 해결안 | -Andriod는 Kotlin을 채팅 기능을 제공  -IOS/mac os는 Swift을 이용해서 채팅 기능을 제공  -Flutter(Dart)을 이용하여 UI 제공 | | | | |
| 승인  기준 | 배꼽 회사 | | | | |
| 제약  사항 |  | | | | |
| 변경  내역 | - 요구사항 추가로 인한 변경 | | | | |
| 관련  부서 | 개발 1팀 | 출처 | |  | |
| 테스트  가능성 | 가능 | 난이도 | | 상 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 요구사항 ID | | 요구사항 명 | | 구분 | | 시스템명 |
| REQ06 | | 올인원 공동 주문 모집 기능 | | 신규 | | 앱 서비스 시스템 |
| 요  구  사  항  내  역 | 설명 | - 주체자가 입점된 음식점 선택 후 주문시 공동 주문 선택 가능.  - 주체자 위치 근방 1km 내외 유저들에게 해당 리스트 표시(20분간 표시)  - 참여자 공동 주문시, 앱에서 원하는 음식 선택 후 결제  - 주체자가 선택한 인원이 다 참여되거나 , 제한시간 넘었을 경우 음식점에  주문 정보 전달. | | | | |
| 해결안 | 1. 개개인들이 직접 UI를 통해, 직접 고른 후 결제.  2. 모두가 결제 된 후, 결제 된 음식을 음식점에게 자동 주문.  3. 주체자가 공동 주문 진행시 자동으로 오픈톡방 생성 참여자들 주문 참여시 해당 오픈톡방 참여. | | | | |
| 승인  기준 | 배꼽 회사 | | | | |
| 제약  사항 |  | | | | |
| 변경  내역 | - 요구사항 추가로 인한 변경 | | | | |
| 관련  부서 | 개발 1팀 | 출처 | |  | |
| 테스트  가능성 | 가능 | 난이도 | | 상 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 요구사항 ID | | 요구사항 명 | | 구분 | | 시스템명 |
| REQ07 | | 개인정보 관리 기능 | | 신규 | | DB 관리 시스템 |
| 요  구  사  항  내  역 | 설명 | - 사용자 개인정보 저장  - 오픈채팅방 ID를 사용하여 저장  - 유저ID로 각각 평가점수 저장  - 사용자 위치 저장  반경 1KM 이내 서비스 이용 하려는 사람들 파악 및 파티구성 | | | | |
| 해결안 | NAVER CLOUD API를 사용해서 현재 위치를 파악하고 파티장과의 거리가 1KM이내에 있다면 참여가능 | | | | |
| 승인  기준 | 팀 내 협의 | | | | |
| 제약  사항 | - 별도의 라이선스나 저작권이 걸린 기술 또는 소프트웨어의 사용은 피한다.  - 다양한 기기에서 동작하도록 시스템 낭비를 줄이는 방향으로 개발한다. | | | | |
| 변경  내역 |  | | | | |
| 관련  부서 | 개발 1팀 | 출처 | |  | |
| 테스트  가능성 | 가능 | 난이도 | | 중 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 요구사항 ID | | 요구사항 명 | | 구분 | | 시스템명 |
| REQ08 | | 음식점 정보 제공 | | 신규 | | DB 관리 시스템 |
| 요  구  사  항  내  역 | 설명 | - 음식점 업체의 위치, 별점, 메뉴, 리뷰 등 정보 제공.  - 음식점 업체의 ID를 사용하여 저장  - 유저ID로 각각 평가점수 저장  - 음식점 업체의 정확한 정보를 사용자에게 제공함으로써 신뢰도를 높임. | | | | |
| 해결안 | 각각의 음식점 업체들 및 브랜드에게 라이선스 획득. | | | | |
| 승인  기준 | 팀 내 협의 | | | | |
| 제약  사항 | - 별도의 라이선스나 저작권이 걸린 기술 또는 소프트웨어의 사용은 피한다.  - 다양한 기기에서 동작하도록 시스템 낭비를 줄이는 방향으로 개발한다. | | | | |
| 변경  내역 |  | | | | |
| 관련  부서 | 개발 1팀 | 출처 | |  | |
| 테스트  가능성 | 가능 | 난이도 | | 상 | |

# 비기능적 요구

## 비기능적 요구사항 목록

| 번호 | 업무구분 | 단위기능 |
| --- | --- | --- |
| REQ91 | 제공 | 사용성 요구사항 – 인터페이스 및 UI 요소 배치 |
| REQ92 | 제공 | 사용성 요구사항 – 도움말 기능 |
| REQ93 | 제공 | 배포 요구사항 - Github |
| REQ94 | 제공 | 구현 요구사항 – Python 언어 사용 |
| REQ95 | 제공 | 안정성 요구사항 – 자료 저장 방식은 DB, DBMS는 MySQL 사용 |

## 비기능적 요구사항 정의

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 요구사항 ID | | 요구사항 명 | | |
| REQ91 | | 사용성 요구사항 – 인터페이스 및 UI 요소 배치 | | |
| 요  구  사  항  내  역 | 설명 | - 사용자가 이용할 때 원활하게 인터페이스를 구현.  - 사용자가 모든 기능에 쉽게 접근할 수 있도록 UI 요소를 배치. | | |
| 해결안 | - 버튼에 알림창이 표시되지 않게 수정. | | |
| 승인  기준 | - 배꼽 회사 | | |
| 제약  사항 |  | | |
| 변경  내역 |  | | |
| 관련  부서 | 개발 2팀 | 출처 |  |
| 테스트  가능성 | 가능 | 난이도 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 요구사항 ID | | 요구사항 명 | | |
| REQ92 | | 사용성 요구사항 - 도움말 기능 | | |
| 요  구  사  항  내  역 | 설명 | - 사용자가 해당 프로그램을 사용하는 데 불편하지 않도록 도움말 제공  - 앱 목적 및 기능 소개  - 앱 이용 방법 제시 | | |
| 해결안 | - 앱에 도움말 탭 추가한 후 해당 탭에서 이용 방법 설명  - md파일을 통해 GitHub에 사용설명서 게시 | | |
| 승인  기준 | 배꼽 회사 | | |
| 제약  사항 |  | | |
| 변경  내역 | - 신규 탭 추가 | | |
| 관련  부서 | 개발 2팀 | 출처 |  |
| 테스트  가능성 | 가능 | 난이도 | 하 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 요구사항 ID | | 요구사항 명 | | |
| REQ93 | | 배포 요구사항 - GitHub | | |
| 요  구  사  항  내  역 | 설명 | - GitHub를 사용함으로써 버전 관리와 팀원 간 협업이 용이하도록 함 | | |
| 해결안 | - 개발팀 GitHub 계정에 소스를 공유하며 프로젝트 진행 | | |
| 승인  기준 | 배꼽 회사 | | |
| 제약  사항 |  | | |
| 변경  내역 |  | | |
| 관련  부서 | 개발 2팀 | 출처 |  |
| 테스트  가능성 | 가능 | 난이도 | 하 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 요구사항 ID | | 요구사항 명 | | |
| REQ94 | | 구현 요구사항 - Python 언어 사용 | | |
| 요  구  사  항  내  역 | 설명 | - 수정과 개발이 용이한 Python를 이용해 개발 | | |
| 해결안 | - Python 이용하여 개발 | | |
| 승인  기준 | 배꼽 회사 | | |
| 제약  사항 |  | | |
| 변경  내역 |  | | |
| 관련  부서 | 개발 1팀 | 출처 |  |
| 테스트  가능성 | 가능 | 난이도 | 중 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 요구사항 ID | | 요구사항 명 | | |
| REQ95 | | 안정성 요구사항 – 자료 저장 방식은 DB, DBMS는 MySQL 사용 | | |
| 요  구  사  항  내  역 | 설명 | - 데이터의 일관성과 무결성, 보안성을 유지하기 위해 데이터베이스 사용  - DBMS는 다양한 운영체제와 여러 가지 프로그래밍 언어를 지원하는 MySQL 사용 | | |
| 해결안 | - MySQL 8.0을 이용하여 개발 | | |
| 승인  기준 | 배꼽 회사 | | |
| 제약  사항 |  | | |
| 변경  내역 |  | | |
| 관련  부서 | 개발 2팀 | 출처 |  |
| 테스트  가능성 | 가능 | 난이도 | 중 |

# 기타 요구 및 제약 사항

## 성능 요구 (반응 시간, 처리 소요 시간, 처리율)

i. 반응 시간

- 1분 이내 같이 배달 시킬 사용자를 찾지 못할 시 알림 발생.

ii. 처리 소요 시간

- 게시물이 만들어지고 5분 이내에 참가자가 모이지 않을 시 사용자에게 알림 발생.

- 서로 안전페이로 결제가 진행이 되고, 한 명 이상이 1분내에 결제를 진행하지 않을 시 결제 취소.

iii. 처리율 : UDP 기준 처리율 70% 예상으로 반응 시간 내 피드백이 없을 시 재전송 구현으로 총 90% 이상의 처리율을 목표.

iv. 동시 접속자 : 1500명 이상.

## H/W 요구 (기억 장치 규모, 통신수용도)

i. 기억 장치 규모 : 500MB 이내

ii. 통신 수용도 : 300 가량 동시 제어 가능

## 예외 조건 및 이의 처리

i. Timeout 에러 : 서버로부터 시간 내에 응답을 얻지 못한 경우 발생.

ii. Notfound 에러 : 웹 페이지나 데이터를 찾을 수 없을 때 발생.

## 자원, 인력에 대한 제약 조건

i. 자원 : 사내 규정에 따른 재정 내에서 개발

ii. 인력 제약 조건 : 사내 규정에 따름

# 인수 조건

## 기능 시험 및 성능 시험

i. 기능 시험 : 반응 속도 성능 만족, 기술한 모든 기능 정상 작동.

ii. 성능 시험 : 10분 간 통합 테스트를 통해 문제가 없을 경우.

# 참고 자료

Android: <https://ko.wikipedia.org/wiki/%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9C_(%EC%9A%B4%EC%98%81_%EC%B2%B4%EC%A0%9C)>

Mysql : <https://ko.wikipedia.org/wiki/MySQL>

Flutter : <https://ko.wikipedia.org/wiki/%ED%94%8C%EB%9F%AC%ED%84%B0>

Swift : https://ko.wikipedia.org/wiki/%EC%8A%A4%EC%9C%84%ED%94%84%ED%8A%B8\_(%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D\_%EC%96%B8%EC%96%B4)

Kotlin : https://ko.wikipedia.org/wiki/%EC%BD%94%ED%8B%80%EB%A6%B0\_(%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D\_%EC%96%B8%EC%96%B4)

Swagger : https://ko.wikipedia.org/wiki/%EC%8A%A4%EC%9B%A8%EA%B1%B0\_(%EC%86%8C%ED%94%84%ED%8A%B8%EC%9B%A8%EC%96%B4)

Python : https://ko.wikipedia.org/wiki/%ED%8C%8C%EC%9D%B4%EC%8D%AC

AWS : https://ko.wikipedia.org/wiki/%EC%95%84%EB%A7%88%EC%A1%B4\_%EC%9B%B9\_%EC%84%9C%EB%B9%84%EC%8A%A4