

# Safe Place 백엔드 개발 포트폴리오

---

김재구

Tel.010-7247-7646

[kimwithglasses@kakao.com](mailto:kimwithglasses@kakao.com)

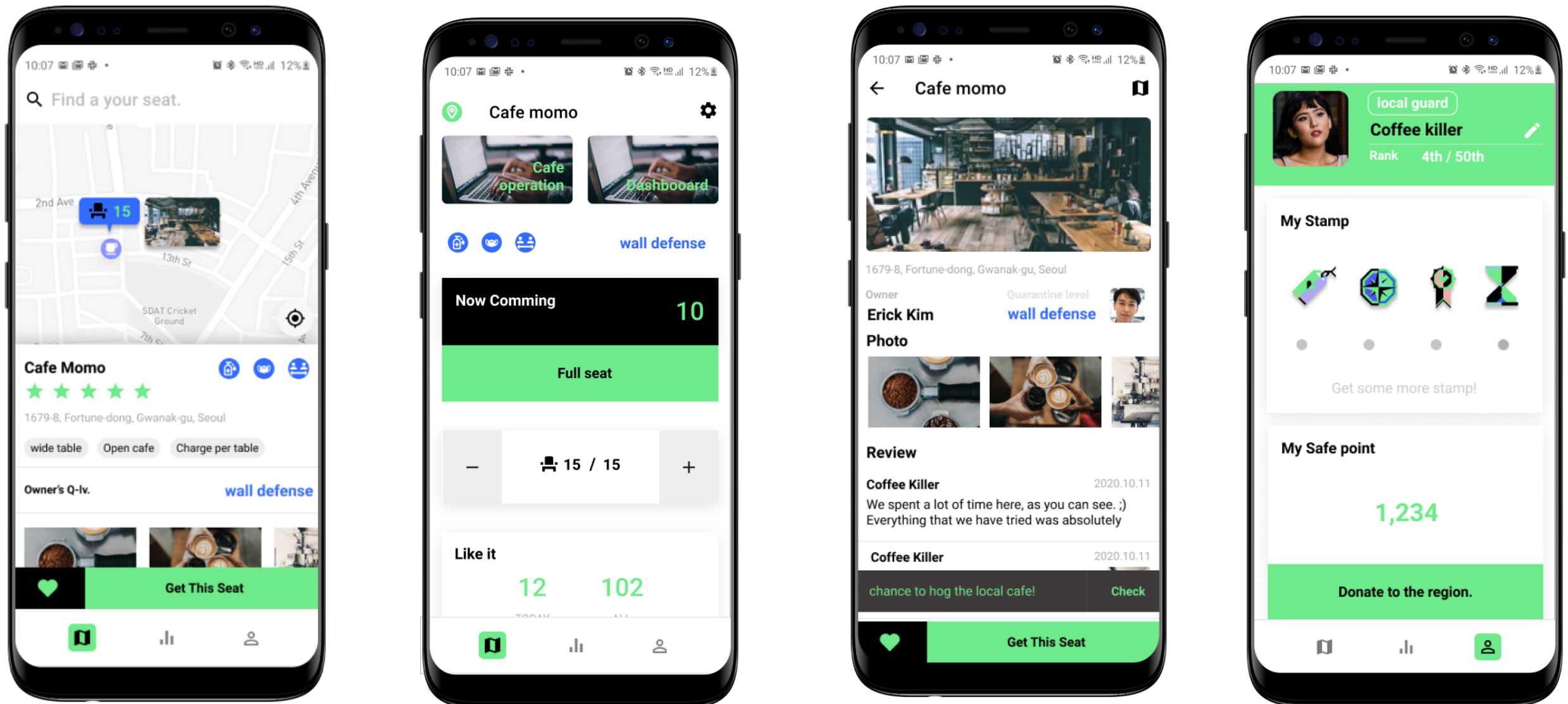
# 목차

---

1. 서비스 소개
2. Github/API 문서 링크
3. Language/Framework
4. 사용된 Spring 버전 및 dependency
5. Authentication - JWT를 이용한 인증구현
6. Unit Test - UML Diagram
7. CI/CD - Workflow

# 서비스 소개

- 코로나-19로부터 안전하게 외출할 수 있도록 사용자의 현재 위치를 기준으로 근처 가게의 혼잡도를 실시간으로 제공해주는 서비스입니다.



# Github/API 링크

---

- **Github :**
  - **<https://github.com/Team-MoBack/Safe-Place-API>**
- **REST API Document**
  - **[https://app.swaggerhub.com/apis-docs/JaeguKim/safe-place rest api/0.0.0/](https://app.swaggerhub.com/apis-docs/JaeguKim/safe-place-rest-api/0.0.0/)**

# Language/Framework

---

Client



**android**



Server



**spring**



# 사용된 Spring 버전 및 dependency

---

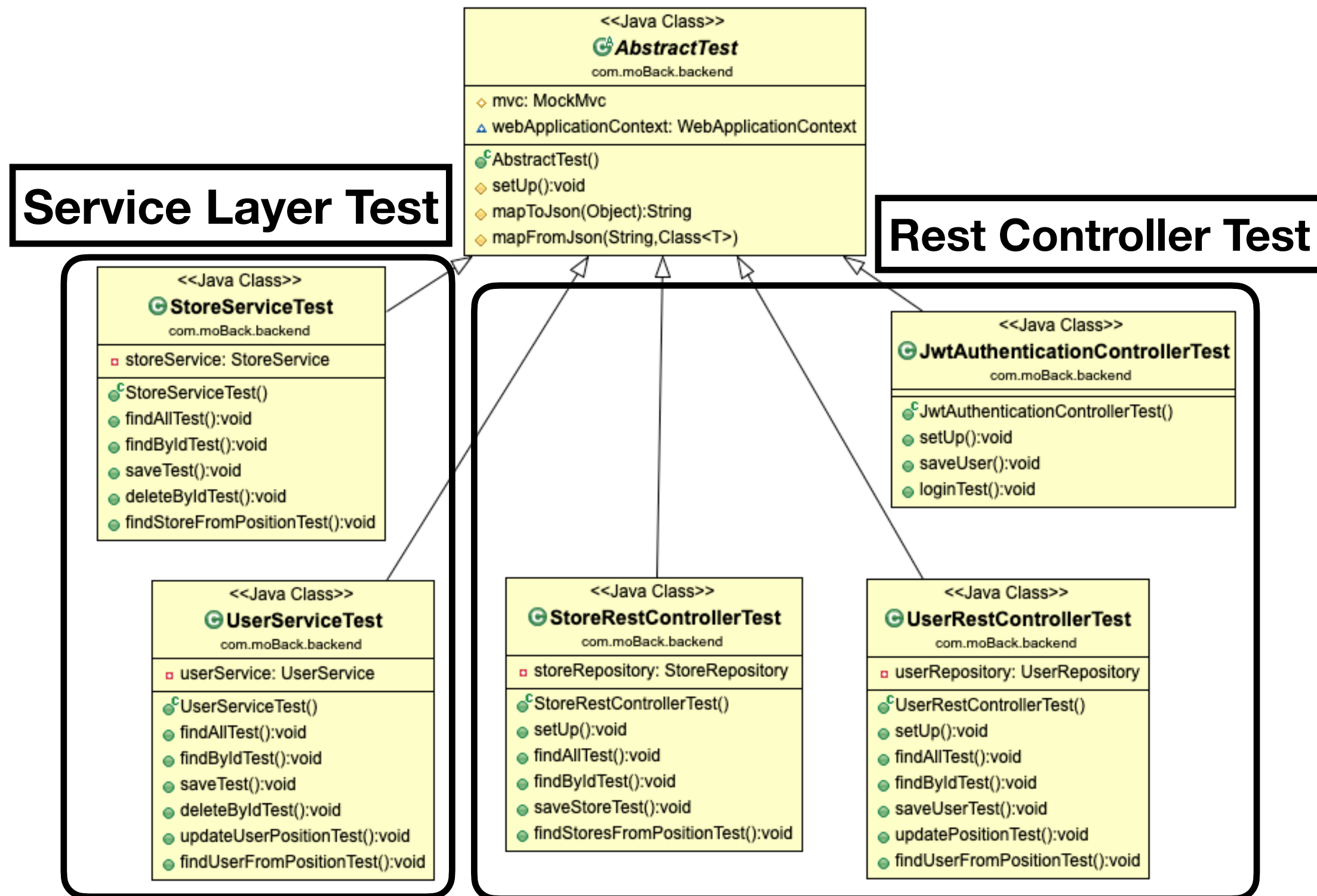
- **Spring Boot - 2.3.4 RELEASE**
- **Docker**
- **Spring Security (JWT)**
- **Swagger2 - 2.9.2**
- **MySQL - 5.1.45**
- **Java 8**
- **JUnit5**
- **Gradle 6.7**

# Authentication - JWT를 이용한 인증구현

- session 정보를 저장할 테이블을 생성할 필요가 없다
- 결과적으로 테이블 공간이 절약된다
- 하지만 그만큼 전송용량이 늘어난다는 단점이 있다.

```
private String doGenerateToken(String id, Map<String, Object> claims) {  
    return Jwts.builder()  
        .setClaims(claims)  
        .setId(id)  
        .setIssuedAt(new Date(System.currentTimeMillis()))  
        .setExpiration(new Date(System.currentTimeMillis() + JWT_TOKEN_VALIDITY * 1000))  
        .signWith(SignatureAlgorithm.HS512, secret)  
        .compact();  
}  
  
public Boolean validateToken(String token, UserDetails userDetails) {  
    final String username = getUsernameFromToken(token);  
    return (username.equals(userDetails.getUsername())) && !isTokenExpired(token);  
}
```

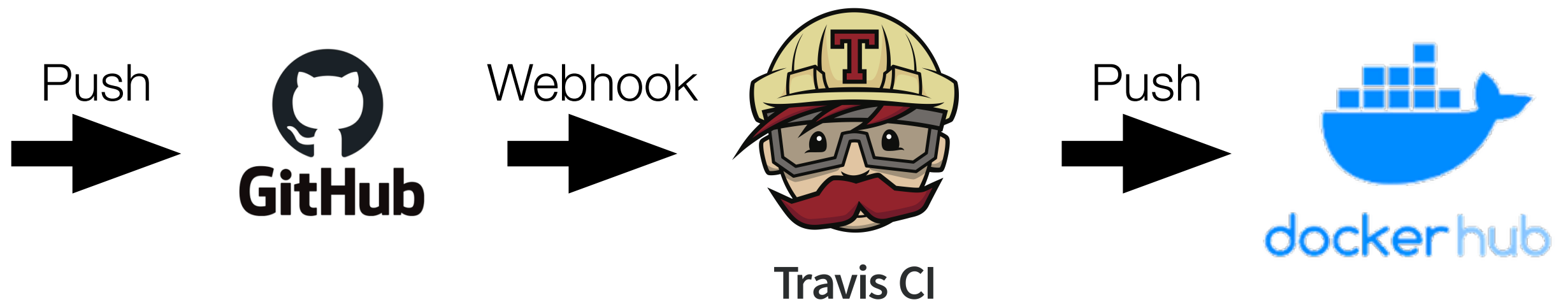
# Unit Test - UML Diagram





# CI/CD - Workflow

---



1. MySQL custom image 빌드, dockerhub에 푸시
2. Spring Boot App Image 빌드,테스트,푸시

# CI/CD - .travis.yml

```
dist: trusty
env:
  global:
    - DOCKER_USER=kimwithglasses
    - secure: AEEIHpGihnGMPyUUF5LosnuoXJiJlRiAhiPpSiXmw+tG2TQRj+092gnm0sZ8mnr0s54WbE7ECorBgDS
jobs:
  include:
    - stage: build safe-place images and push to dockerhub repository
      script:
        - echo "$DOCKER_PASS" | docker login -u "$DOCKER_USER" --password-stdin
        - docker build --tag kimwithglasses/safe-place-db:0.0.1 . && docker push kimwithglasses
        - cd SafePlaceAPI && ./gradlew clean build -Pprofile=ci -x test && docker build --tag k
          . && docker push kimwithglasses/safe-place-api
```

커스텀 MySQL 이미지 빌드 후 Push

ci profile로 Spring Boot App 빌드 및 Push

# CI/CD - 서버실행

```
docker-compose.yml
version: "3"
services:
  safe-place-db:
    container_name: safe-place-db
    image: kimwithglasses/safe-place-db:0.0.1
    ports:
      - 3305:3306

  safe-place-api:
    container_name: safe-place-api
    image: kimwithglasses/safe-place-api:0.0.1
    restart: on-failure:10
    ports:
      - 80:8080
    depends_on:
      - safe-place-db
```

Host의 환경에  
관계없이 실행가능

DB 서버에서  
세팅(테이블 생성)  
이 완료될때까지  
최대 10번 실행