

# Careville

---

청소 서비스 관리 시스템

시스템 구조 및 운영 매뉴얼

Version 1.0.0

2024년 12월



# 목차

---

## 1. 시스템 개요

- 1.1 소개
- 1.2 주요 기능
- 1.3 기술 스택

## 2. 시스템 아키텍처

- 2.1 전체 구조
- 2.2 레이어 아키텍처
- 2.3 데이터 흐름

## 3. 비즈니스 모듈

- 3.1 상담 관리
- 3.2 사례 관리
- 3.3 게시판 관리
- 3.4 리뷰 관리
- 3.5 카테고리 관리
- 3.6 상품 관리

## 4. 데이터베이스 구조

- 4.1 스키마 설계
- 4.2 주요 테이블
- 4.3 테이블 관계

## 5. API 문서

- 5.1 관리자 API
- 5.2 공개 API
- 5.3 보안 및 인증

## 6. 운영 가이드

- 6.1 설치 및 실행
- 6.2 환경 설정
- 6.3 모니터링

## 7. 보안 고려사항

## 8. 부록

# 1. 시스템 개요

## 1.1 소개

Careville 청소 서비스 관리 시스템은 청소 서비스 업체의 효율적인 운영을 위한 통합 관리 플랫폼입니다. 고객 상담부터 서비스 제공, 사후 관리까지 전 과정을 체계적으로 관리할 수 있도록 설계되었습니다.

### 핵심 가치:

- 고객 상담 및 문의 통합 관리
- 서비스 포트폴리오 체계적 관리
- 실시간 고객 피드백 수집 및 분석
- 효율적인 관리자 운영 시스템

## 1.2 주요 기능

### 1.2.1 고객 관리 기능

- **상담 관리:** 고객 문의 접수, 처리 상태 추적, 답변 관리
- **리뷰 관리:** 고객 후기 수집, 평점 관리, 비밀번호 보호 기능
- **파일 관리:** 상담 관련 파일 업로드 및 관리

### 1.2.2 콘텐츠 관리 기능

- **사례 관리:** 서비스 포트폴리오 등록 및 관리
- **게시판 관리:** 공지사항, FAQ, 포토 갤러리 운영
- **카테고리 관리:** 서비스 분류 체계 관리
- **상품 관리:** 서비스 상품 등록, 가격 관리, 할인을 설정

### 1.2.3 시스템 관리 기능

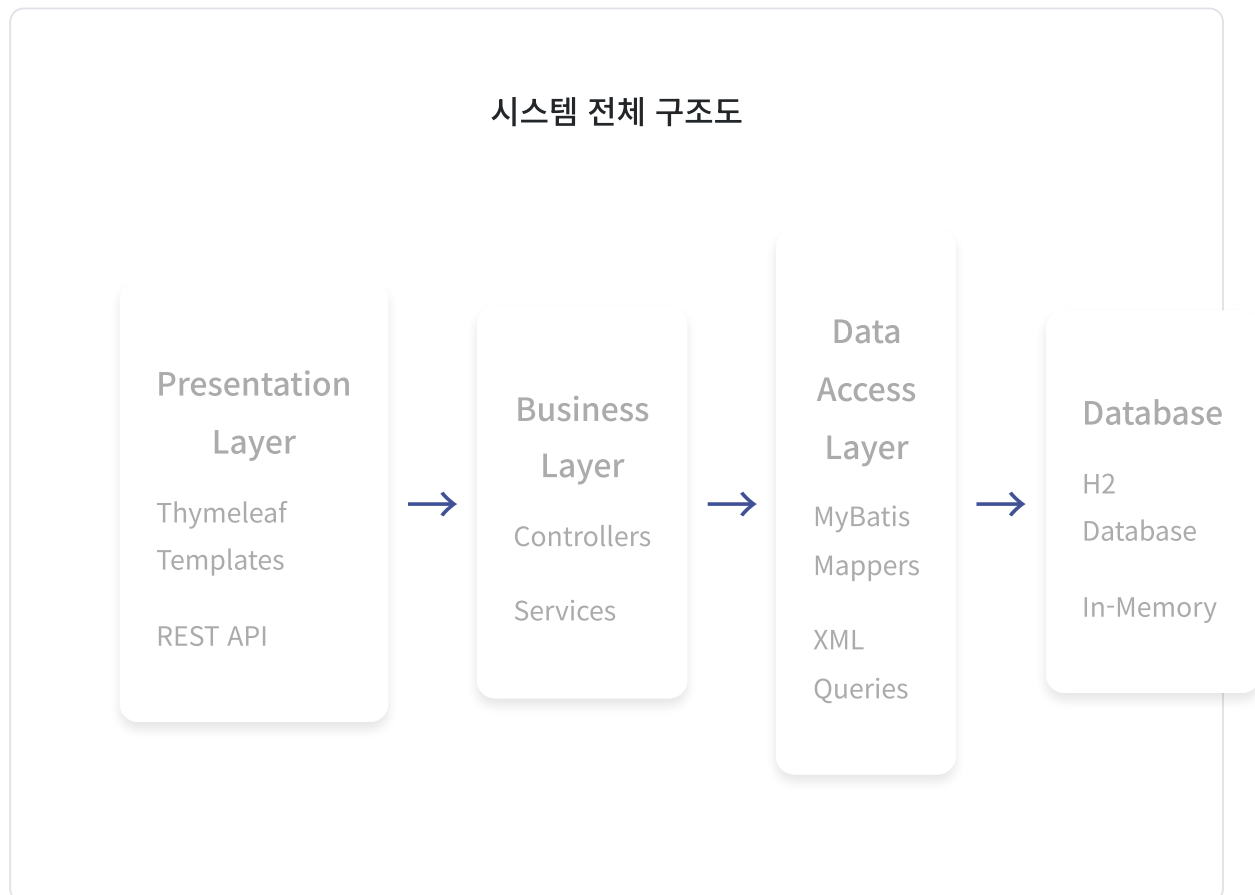
- **관리자 인증:** 세션 기반 보안 인증
- **공통 코드 관리:** 시스템 전반 코드 통합 관리
- **API 제공:** RESTful API를 통한 외부 연동 지원

## 1.3 기술 스택

구분	기술	버전	용도
Backend Framework	Spring Boot	3.5.4	웹 애플리케이션 프레임워크
Language	Java	17	프로그래밍 언어
Database	H2 Database	2.2.224	인메모리 데이터베이스
ORM	MyBatis	3.0.3	데이터베이스 매핑
Template Engine	Thymeleaf	3.x	서버사이드 템플릿
API Documentation	Swagger/OpenAPI	2.2.0	API 문서화
Security	BCrypt, AES	-	암호화 및 보안
Build Tool	Maven	3.x	빌드 및 의존성 관리

## 2. 시스템 아키텍처

### 2.1 전체 구조



### 2.2 레이어 아키텍처

#### 2.2.1 Controller Layer

HTTP 요청을 처리하고 적절한 서비스로 라우팅합니다.

Controller	경로 패턴	담당 기능
CnsltCtrl	/apage/cnslt0*.do	상담 관리
CaseCtrl	/apage/case0*.do	사례 관리
BoardCtrl	/apage/board0*.do	게시판 관리
ReviewCtrl	/apage/review0*.do, /api/reviews	리뷰 관리
CategoryCtrl	/apage/category0*.do	카테고리 관리
ProductCtrl	/apage/product0*.do	상품 관리
ProductApiController	/api/v1/category-products.do	상품 API

#### URL 패턴 규칙:

- 010: 목록 조회
- 020: 상세 조회
- 030: 등록
- 040: 수정
- 051: 삭제

## 2.2.2 Service Layer

비즈니스 로직을 처리하고 트랜잭션을 관리합니다.

- **CnsltSvc:** 상담 비즈니스 로직
- **CaseSvc:** 사례 비즈니스 로직
- **BoardSvc:** 게시판 비즈니스 로직
- **ReviewSvc:** 리뷰 비즈니스 로직, 비밀번호 검증
- **CategorySvc:** 카테고리 비즈니스 로직
- **ProductSvc:** 상품 비즈니스 로직
- **CmmnSvc:** 공통 기능 (파일 처리, 코드 관리)



## 2.2.3 Data Access Layer

MyBatis를 통한 데이터베이스 접근을 담당합니다.

```
// Mapper Interface 예시
@Mapper
public interface ReviewMapper {
    List<HashMap<String, Object>> getReviewList(HashMap<String, Object> paramMap);
    HashMap<String, Object> getReviewView(HashMap<String, Object> paramMap);
    int setReviewReg(HashMap<String, Object> paramMap);
    int setReviewUpd(HashMap<String, Object> paramMap);
    int setReviewDel(HashMap<String, Object> paramMap);
}
```

## 2.3 데이터 흐름

1. **요청 수신:** 클라이언트로부터 HTTP 요청 수신
2. **Controller 처리:** 요청을 적절한 Controller 메소드로 라우팅
3. **Service 호출:** 비즈니스 로직 실행
4. **Mapper 실행:** 데이터베이스 쿼리 수행
5. **응답 생성:** View(Thymeleaf) 또는 JSON 응답 반환

## 3. 비즈니스 모듈

### 3.1 상담 관리

#### 3.1.1 기능 개요

고객의 청소 서비스 문의를 접수하고 관리하는 핵심 모듈입니다.

#### 3.1.2 주요 기능

- 상담 접수 및 등록
- 상담 상태 관리 (접수, 진행중, 완료)
- 관리자 답변 작성 및 관리
- 파일 첨부 기능
- 상담 이력 조회

#### 3.1.3 데이터 구조

필드명	타입	설명
CNSLT_SEQ	INT	상담 고유번호
NM	VARCHAR(30)	고객명
TEL1, TEL2	VARCHAR(20)	연락처
SVC_CN_CD	VARCHAR(3)	서비스 내용 코드
INQRY_CN	CLOB	문의 내용
ANSWER	CLOB	답변 내용
STTUS_CD	VARCHAR(3)	진행 상태

## 3.2 사례 관리

### 3.2.1 기능 개요

청소 서비스 포트폴리오를 관리하여 고객에게 서비스 품질을 시각적으로 전달합니다.

### 3.2.2 주요 기능

- 사례 등록 및 수정
- 대표 이미지 관리
- Before/After 이미지 비교
- 카테고리별 분류

## 3.3 게시판 관리

### 3.3.1 기능 개요

공지사항, FAQ, 포토갤러리 등 다양한 형태의 게시판을 통합 관리합니다.

### 3.3.2 게시판 유형

- **공지사항:** 서비스 관련 공지 및 안내
- **FAQ:** 자주 묻는 질문과 답변
- **포토갤러리:** 서비스 사진 공유

### 3.3.3 특수 기능

- 비밀번호 설정 (비밀번호 보호)
- 공지글 상단 고정
- 파일 첨부 지원

## 3.4 리뷰 관리

### 3.4.1 기능 개요

고객의 서비스 이용 후기를 수집하고 관리하는 모듈입니다.

### 3.4.2 주요 기능

- 리뷰 작성 및 수정
- 별점 평가 (1-5점)
- 비밀번호 보호 기능
- 서비스별 리뷰 분류
- 노출 여부 관리

### 3.4.3 보안 기능

#### 비밀번호 검증 프로세스:

- BCrypt 해싱을 통한 비밀번호 암호화
- 비밀번호 미제공 시: HTTP 401 Unauthorized
- 잘못된 비밀번호: HTTP 403 Forbidden
- 올바른 비밀번호: HTTP 200 OK

## 3.5 카테고리 관리

### 3.5.1 기능 개요

서비스 분류 체계를 관리하여 상품과 서비스를 체계적으로 구성합니다.

### 3.5.2 주요 기능

- 카테고리 생성/수정/삭제
- 정렬 순서 관리
- 사용 여부 설정

- 카테고리별 상품 연결

## 3.6 상품 관리

### 3.6.1 기능 개요

청소 서비스 상품을 등록하고 가격 정책을 관리합니다.

### 3.6.2 주요 기능

- 상품 정보 관리 (이름, 설명, 평형대)
- 가격 설정 (정가, 판매가, 할인율)
- 서비스 소요시간 관리
- 노출 순서 및 상태 관리
- 카테고리별 상품 분류

### 3.6.3 가격 정책

항목	설명
정가	기본 서비스 가격
판매가	실제 판매 가격
할인율	정가 대비 할인 비율 (%)

## 4. 데이터베이스 구조

### 4.1 스키마 설계

시스템은 H2 인메모리 데이터베이스를 사용하며, 애플리케이션 시작 시 스키마가 자동으로 생성됩니다.

### 4.2 주요 테이블

#### 4.2.1 TB\_CONSULTATION (상담)

컬럼명	타입	제약조건	설명
CNSLT_SEQ	INT	PK, AUTO_INCREMENT	상담 일련번호
NM	VARCHAR(30)	NOT NULL	고객명
TEL2	VARCHAR(20)	NOT NULL	휴대전화
INQRY_CN	CLOB	NOT NULL	문의내용

#### 4.2.2 TB\_REVIEW (리뷰)

컬럼명	타입	제약조건	설명
REVIEW_SEQ	INT	PK, AUTO_INCREMENT	리뷰 일련번호
REVIEW_NM	VARCHAR(50)	NOT NULL	작성자명
STAR_RATE	INT	NOT NULL	별점 (1-5)
PW	VARCHAR(99)	NULL	비밀번호 (BCrypt)

### 4.2.3 TB\_CATEGORY (카테고리)

컬럼명	타입	제약조건	설명
SERVICE_CD	VARCHAR(3)	PK	서비스 코드
SERVICE_NM	VARCHAR(100)	NOT NULL	서비스명
SERVICE_ORDER	INT	DEFAULT 0	정렬 순서
USE_YN	VARCHAR(1)	DEFAULT 'Y'	사용 여부

### 4.2.4 TB\_PRODUCT (상품)

컬럼명	타입	제약조건	설명
PRODUCT_NO	INT	PK, AUTO_INCREMENT	상품번호
SERVICE_CD	VARCHAR(3)	FK, NOT NULL	서비스 코드
PRODUCT_NM	VARCHAR(200)	NOT NULL	상품명
SALE_PRICE	INT	DEFAULT 0	판매가

## 4.3 테이블 관계

#### 주요 관계:

- TB\_PRODUCT.SERVICE\_CD → TB\_CATEGORY.SERVICE\_CD (외래키)
- TB\_FILE\_RELATION → TB\_FILE (파일 연결)
- TB\_FILE\_RELATION → 각 비즈니스 테이블 (다대다 관계)
- TB\_CMMN\_CODE: 독립적인 코드 테이블

## 5. API 문서

### 5.1 관리자 API

관리자 API는 세션 인증이 필요하며, 주로 관리 화면에서 사용됩니다.

#### 5.1.1 상담 관리 API

POST /cnsltList.do

상담 목록 조회

POST /cnsltView.do

상담 상세 조회

POST /cnsltUpd.do

상담 답변 수정

#### 5.1.2 리뷰 관리 API

POST /reviewList.do

리뷰 목록 조회

POST /reviewReg.do

리뷰 등록

POST /reviewUpd.do



리뷰 수정

POST /reviewDel.do

리뷰 삭제

## 5.2 공개 API

공개 API는 인증 없이 접근 가능하며, 프론트엔드 애플리케이션에서 직접 호출할 수 있습니다.

### 5.2.1 리뷰 API

GET /api/reviews

리뷰 목록 조회 (페이징)

파라미터	타입	필수	설명
pageNum	int	N	페이지 번호 (기본값: 1)
pageSize	int	N	페이지 크기 (기본값: 10)
starRate	int	N	별점 필터 (1-5)
dispYn	string	N	노출 여부 (Y/N)

GET /api/reviews/{reviewSeq}

리뷰 상세 조회

파라미터	타입	필수	설명
reviewSeq	int	Y	리뷰 일련번호
password	string	N	비밀번호 (보호된 리뷰의 경우)

POST /api/reviews

리뷰 등록

```
{
  "reviewNm": "홍길동",
  "reviewCn": "서비스가 매우 만족스러웠습니다.",
  "starRate": 5,
  "svcCnCd": "001",
  "svcDate": "2024-01-15",
  "pw": "password123", // 선택사항
  "dispYn": "Y"
}
```

POST /api/reviews/{reviewSeq}

리뷰 수정

DELETE /api/reviews/{reviewSeq}

리뷰 삭제

## 5.2.2 상품 카테고리 API

GET /api/v1/category-products.do

카테고리 및 상품 통합 조회

파라미터	타입	필수	설명
groupByCategory	boolean	N	카테고리별 그룹핑 여부
includeStats	boolean	N	통계 정보 포함 여부
onlyActive	boolean	N	활성 상품만 조회

## 5.3 보안 및 인증

### 5.3.1 세션 인증

관리자 API는 세션 기반 인증을 사용합니다.

- 세션 키: `LCMS_SESSION`
- 로그인 상태: `KF_LOGON`
- 세션 타임아웃: 30분 (설정 가능)

### 5.3.2 비밀번호 보호

리뷰 및 게시글에 대한 비밀번호 보호 기능:

- BCrypt 해싱 알고리즘 사용
- Salt 자동 생성
- Cost Factor: 10 (기본값)

### 5.3.3 CORS 설정

```
@CrossOrigin(origins = {  
    "http://localhost:3000",  
    "http://localhost:3001",  
    "http://localhost:8080"  
})
```

## 6. 운영 가이드

---

### 6.1 설치 및 실행

#### 6.1.1 시스템 요구사항

- Java 17 이상
- Maven 3.x
- 최소 메모리: 512MB
- 권장 메모리: 1GB 이상

#### 6.1.2 빌드 및 실행

##### Windows 환경

```
# 프로젝트 빌드
mvnw.cmd clean package

# 애플리케이션 실행
mvnw.cmd spring-boot:run
```

##### Linux/Mac 환경

```
# 프로젝트 빌드
./mvnw clean package

# 애플리케이션 실행
./mvnw spring-boot:run
```

##### JAR 파일 직접 실행

```
java -jar target/cleaning-0.0.1-SNAPSHOT.war
```

### 6.1.3 접속 URL

- 관리자 패널: <http://localhost:8081/apage/home.do>
- API 문서: <http://localhost:8081/swagger-ui.html>
- H2 콘솔: <http://localhost:8081/h2-console>

## 6.2 환경 설정

### 6.2.1 application.yml 주요 설정

```
server:
  port: 8081 # 서버 포트

spring:
  datasource:
    url: jdbc:log4jdbcp:h2:mem:testdb # H2 인메모리 DB
    username: sa
    password:

  h2:
    console:
      enabled: true # H2 콘솔 활성화

  kframe:
    viewRowCnt: 10 # 페이지당 행 수
  aes:
    key: my-secret-cleaning-12345 # AES 암호화 키
    filePath: C:/Users/nun/Downloads/careville # 파일 업로드 경로
```

### 6.2.2 데이터베이스 설정

#### H2 Database 특징:

- 인메모리 모드로 실행 (재시작 시 데이터 초기화)
- 개발 및 테스트 환경에 적합
- 운영 환경에서는 영구 저장소 모드 권장

## 6.2.3 로깅 설정

```
logging:
  level:
    root: INFO
    kr.co.cleaning: DEBUG # 애플리케이션 로그
    jdbc.sqltiming: DEBUG # SQL 로그
```

## 6.3 모니터링

### 6.3.1 로그 모니터링

- 애플리케이션 로그: `logs/application.log`
- SQL 로그: 콘솔 출력 또는 설정된 파일
- 에러 로그: ERROR 레벨 이상 별도 기록

### 6.3.2 성능 모니터링

- JVM 메모리 사용량 모니터링
- HTTP 요청 응답 시간 측정
- 데이터베이스 커넥션 풀 상태 확인

### 6.3.3 헬스 체크

시스템 상태 확인을 위한 엔드포인트:

- 애플리케이션 상태 확인
- 데이터베이스 연결 상태
- 디스크 공간 확인

## 7. 보안 고려사항

---

### 7.1 인증 및 권한

#### 7.1.1 세션 관리

- 세션 타임아웃 설정 (기본 30분)
- 세션 고정 공격 방지
- 동시 세션 제한

#### 7.1.2 비밀번호 정책

- BCrypt 해싱 (Cost Factor: 10)
- 최소 길이: 8자
- 복잡도 요구사항 적용 권장

### 7.2 데이터 보호

#### 7.2.1 암호화

- **AES 암호화:** 민감한 데이터 암호화
- **BCrypt:** 비밀번호 해싱
- **HTTPS:** 전송 구간 암호화 (운영 환경)

#### 7.2.2 입력 검증

- SQL Injection 방지 (PreparedStatement 사용)
- XSS 방지 (입력값 이스케이프)
- 파일 업로드 검증 (확장자, 크기 제한)

## 7.3 접근 제어

### 7.3.1 API 접근 제어

- 관리자 API: 세션 인증 필수
- 공개 API: CORS 정책 적용
- Rate Limiting 적용 권장

### 7.3.2 파일 접근 제어

- 업로드 디렉토리 외부 접근 차단
- 실행 파일 업로드 금지
- 파일 크기 제한 (10MB/파일, 30MB/요청)

#### 보안 체크리스트:

- ✓ 운영 환경에서 H2 콘솔 비활성화
- ✓ 기본 비밀번호 변경
- ✓ HTTPS 적용
- ✓ 정기적인 보안 패치
- ✓ 로그 모니터링 설정



# 8. 부록

## 8.1 트러블슈팅

### 8.1.1 자주 발생하는 문제

문제	원인	해결방법
포트 충돌	8081 포트 사용 중	application.yml에서 포트 변경
메모리 부족	JVM 힙 메모리 부족	-Xmx 옵션으로 메모리 증가
파일 업로드 실패	디렉토리 권한 문제	업로드 디렉토리 권한 확인
H2 연결 실패	잘못된 JDBC URL	jdbc:h2:mem:testdb 확인

## 8.2 성능 최적화

### 8.2.1 데이터베이스 최적화

- 적절한 인덱스 생성
- 쿼리 최적화 (N+1 문제 해결)
- 커넥션 풀 크기 조정

### 8.2.2 애플리케이션 최적화

- 캐싱 전략 수립
- 비동기 처리 활용
- 정적 리소스 CDN 활용

## 8.3 참조 링크

- **Spring Boot 공식 문서:** <https://spring.io/projects/spring-boot>
- **MyBatis 문서:** <https://mybatis.org/mybatis-3/ko/>
- **Thymeleaf 문서:** <https://www.thymeleaf.org/>
- **H2 Database:** <http://www.h2database.com/>

## 8.4 용어집

용어	설명
CNSLT	Consultation (상담)
SVC	Service (서비스)
MNGR	Manager (관리자)
CMMN	Common (공통)
REG	Register (등록)
UPD	Update (수정)
DEL	Delete (삭제)

## Careville 청소 서비스 관리 시스템

Version 1.0.0

© 2024 Careville. All rights reserved.