

# 2023년 구디 아카데미 HCM Final 프로젝트

[IT개발회사 그룹웨어 구축]

---

[HCM\_PM\_001]

개발표준정의서

Ver. 1.0

## Document Information

프로젝트	그룹웨어
단계	설계
활동	애플리케이션 설계
작업	개발 표준 정의
산출물	개발 표준 정의서
파일명	HCM_001_개발표준정의서.doc

## Revision History

날짜	버전	작성 / 변경 / 추가 내역	요청자	승인자
2024. 02. 12	1.0	문서 최초 작성	오지수	오지수

# Contents

Document Information	2
Revision History	2
<b>1. Development Configuration</b>	<b>5</b>
<b>2. Configuration Management Plan</b>	<b>7</b>
<b>3. Naming Rules</b>	<b>8</b>
3.1. Standard Naming rule	8
3.2. Dictionary of Terms	8
3.3. Java	10
3.3.1. Package	10
3.3.2. Class	11
3.3.3. Interface	11
3.3.4. Method	11
3.4. JSP	12
3.5. Javascript	13
3.6. CSS	13
<b>4. Database</b>	<b>14</b>
<b>5. API Annotation Process</b>	<b>16</b>
5.1. Class	16
5.2. Method	16
<b>6. Log Process</b>	<b>17</b>
6.1. Basic Rules	17
6.2. Log Level Rules	17
6.3. User Log Processing Rules	17

<b>7. Test Case Creation Rules</b>	19
<b>8. Directory Structure</b>	21
8.1. src/main/java	21
8.2. src/main/resources	21
8.3. src/main/webapp	22
8.4. src/test/java	22

# 1. Development Configuration

개발 도구	Eclipse 2022-03	미들웨어	Tomcat 9.0.83
	DBeaver		Oracle 21c XE
개발 언어	jdk 17.0.9	Framework	Redis
	Javascript		Spring Framework 5.3.20
	JSTL	라이브러리	MyBatis 3.5.15
	Expression Language		jQuery 3.7.0
	Oracle SQL		jQuery-ui
UI	Bootstrap		spring-context 5.3.20
	HTML		spring-webmvc 5.3.20
	CSS		spring-orm 5.3.20
Platform	Oracle Cloud		slf4j-api 2.0.7
	Docker		jcl-over-slf4j 2.0.7
API	jsTree 3.2.1		logback-core 1.4.14
	Redis 2.6.0		logback-classic 1.4.14
	datatables 1.13.7		jstl 1.2
	commons-fileupload 1.5		ojdbc10 19.21.0.0
	commons-io 2.11.0		mybatis 3.5.9
	dropzone		mybatis-spring 3.0.3
	signature-pad 1.5.3		HikariCP 4.0.3
	Html2pdf 1.3.4		lombok 1.18.30
	fullcalendar 5.7.0		aspectjrt 1.9.21
	DateTimePicker		aspectjweaver 1.9.21
	Google Calendar		gson 2.10.1
	Cron Schedule		spring-security 5.7.1

			spring-data-redis 2.6.0
			spring-session-data-redis 2.6.0
			lettuce-core 6.1.5.RELEASE
테스트 도구	Postman		
	spring-test 5.3.20		
	junit 4.12		

## 2. Configuration Management Plan

1. Git을 사용하여 단위 개발 및 Version 관리를 하며, 무료 깃 저장소 호스팅 서비스 GitHub를 사용한다.
2. 개발자는 개인의 branch로만 push 할 수 있으며, 하루에 최소 1번은 push한다. push는 매일 17시 전에 완료한다.
3. 형상 관리자는 매일 17시 이후에 merge 작업을 수행하며, 개발자는 형상관리자가 conflict 해결 등 모든 작업이 수행한 이후에 다시 push 할 수 있다.
4. 개발 진행 중인 작업 내용의 경우 개인 단위의 백업 관리가 이루어지며, 개발 완료 작업 내용과 일(daily) 단위의 백업은 운영자가 맡는다.
5. 공통 파일 수정 및 파일 이름 변경이 필요할 경우 운영자(부재중일시 형상 관리자)에게 요청해야 하며, 공통 파일 수정 및 파일 이름 변경은 운영자(부재중일시 형상 관리자)만 가능하다.

### 3. Naming Rules

#### 3.1. Standard Naming rule

명명법	설명	적용
<b>pascal</b>	대문자로 시작하며 의미 있는 단어의 첫 글자는 대문자로 한다.	클래스, 파일
<b>camel</b>	소문자로 시작하며 의미 있는 단어의 첫 글자는 대문자로 한다.	변수, 메소드
<b>upper</b>	대문자로만 작성한다.	상수
<b>lower</b>	소문자로만 작성한다.	예약어, 패키지
<b>under snake</b>	대문자로만 작성하며 단어 연결은 언더바(_)로 한다.	DB테이블

#### 3.2. Dictionary of Terms

설명	용어	설명	설명
Employee	사원	File	파일
Department	부서	List	목록
User	사용자	Detail	상세
Admin	관리자	Comm	공통
Resource	자원	Popup	팝업
Reservation	예약	Login	로그인
Schedule	일정	Logout	로그아웃
Sign	결재	Approval	승인
Mobile	휴대폰	Refuse	반려
Board	게시판	Regist	등록



Document	문서	Search	검색
Title	게시글 제목	Update	수정
Content	게시글 내용	Cancel	취소
Writer	작성자	Delete	삭제
RegDate	작성일	Restore	복원
Reply	답글	Status	상태
Page	페이지	Template	템플릿

용어사전 외 명사는 한국어 낱말을 로마자 표기법에 따라 영문으로 변경하여 사용하며, 한글 로마자 표기는 아래표를 참고하여 작성한다.

순번	자음	초성	종성
1	ㄱ	g	k
2	ㄲ	kk	kk
3	ㅋ	c	k
4	ㄷ	d	t
5	ㄸ	dd	
6	ㅌ	t	t
7	ㅂ	b	p
8	ㅃ	bb	
9	ㅍ	p	p
10	ㅈ	j	j
11	ㅉ	jj	
12	ㅊ	ch	ch
13	ㅅ	s	s
14	ㅆ	ss	ss

순번	모음	중성
1	ㅏ	a
2	ㅑ	eo
3	ㅓ	o
4	ㅕ	u
5	ㅡ	eu
6	ㅣ	i
7	ㅗ	ae
8	ㅛ	e
9	ㅜ	oi
10	ㅠ	ya
11	ㅡ	yeo
12	ㅝ	yo
13	ㅠ	yu
14	ㅞ	yae

15	ㅎ	h	h
16	ㅁ	m	m
17	ㄴ	n	n
18	ㅇ		ng
19	ㄹ	r	l

15	예	ye
16	와	wa
17	왜	wae
18	wo	wo
19	we	we
20	에이	eui
21	위	wi

### 3.3. Java

#### 3.3.1. Package

위치	패키지명	설명
src/main/Java	com.hcm.grw	base package
	com.hcm.grw.ctrl	controller 파일 정의
	com.hcm.grw.aop	Aop 파일 정의
	com.hcm.grw.comm	공통 모듈
	com.hcm.grw.config	API Configuration 파일 정의
	com.hcm.grw.filter	Filter 파일 정의
	com.hcm.grw.interceptor	Interceptor 파일 정의
	com.hcm.grw.model.mapper	Dao 파일 정의
	com.hcm.grw.model.service	Service 파일 정의
	com.hcm.grw.socket	WebSocket 파일 정의
	com.hcm.grw.dto	Dto(vo) 파일 정의

<b>src/main/resource</b>	config	mybatis configuration 정의
	message	Message 소스 정의
	properties	oracle 접속 정보 정의
	sqls	Mapper 정의

### 3.3.2. Class

suffix	설명
<b>Ctrl</b>	Controller Class File
<b>Aop</b>	Aop Class File
<b>Filter</b>	Filter Class File
<b>DaoImpl</b>	Dao Class File
<b>ServiceImpl</b>	Service Class File
<b>Dto</b>	Dto(vo) File
<b>JUnitTest</b>	JUnit Test File

### 3.3.3. Interface

suffix	설명
<b>Dao</b>	Dao Interface File
<b>Service</b>	Service Interface File

### 3.3.4. Method

구분	유형	prefix	비고
data 처리	등록 / 입력	insert	
	조회	select	
	다중 조회	select	suffix로 List 사용
	수정	update	delflag 변경 포함
	삭제	delete	
	다중 등록/수정/삭제	각 prefix	suffix로 Multi 사용
Value Object	값 읽기	Get	
	값 설정	set	

### 3.4. JSP

1. 파일명은 [용어사전].jsp로 한다.
2. jsp의 각 컨트롤의 변수명은 용어사전과 아래 표를 적절히 조합하여 사용한다.

컨트롤	prefix
button	btn
image	img
text	txt
textArea	txa
select	sel
listBox	lst
radio	rdo
checkbox	chk

div	box
table	tt

### 3.5. Javascript

1. 파일명은 import 하는 jsp 파일의 이름을 따른다.
2. function명은 under snake 방식에 따라 fn\_[동사]\_[명사]() 로 하며, 동사 및 명사는 용어사전을 기준으로 한다.
3. 변수명은 camel 방식에 따라 용어사전을 적절히 조합하여 사용한다.

### 3.6. CSS

1. 파일명은 import 하는 jsp 파일의 이름을 따른다.

## 4. Database

구분	용어	약어
객체	Table	TT
	View	VW
	Sequence	SEQ
	Trigger	TRI
	Function	FN
	Procedure	PRO
	Index	IDX
제약조건	Primary Key	PK
	Foreign Key	FK
	Unique Key	UK
	Not Null	NN
	Check	CHK
	Default	DF
Trigger 수행어	Insert	I
	Update	U
	Delete	D
	Before	B
	After	A

1. 객체의 이름에 객체의 약어를 **suffix**로 사용한다. **Table**의 경우에는 사용하지 않는다.
2. 의미있는 단어는 ‘\_’로 연결하며, 3단위 이상을 넘어가지 않도록 한다.
3. 테이블의 컬럼명에는 해당 테이블이 1단위일 때는 1단위 글자의 앞 4자리, 2단위일 때는 각 단위

글자의 앞 2자리, 3단위 일 때는 1단위 글자의 앞 2자리, 나머지 각 단위 글자의 앞 1자리를 **prefix**로 사용한다.

## 5. API Annotation Process

### 5.1. Class

```
/**  
  
* 클래스 상세설명  
  
* @author : 담당자  
  
* @since : 생성일자(yyyy. mm. dd.)  
  
* @version : 버전  
  
*/
```

### 5.2. Method

```
/**  
  
* 기능 상세설명  
  
* @param : 전달 받은 parameter 설명(변수)/타입  
  
* @return : 처리 후 반환되는 설명(변수)/타입  
  
* @author : 담당자  
  
* @since : 생성일자(yyyy. mm. dd.)  
  
*/
```



## 6. Log Process

### 6.1. Basic Rules

로그 처리는 logback 라이브러리 및 slf4j를 사용한다.

### 6.2. Log Level Rules

로그 레벨	설명	사용 예시
DEBUG	디버깅 용 상세 정보들	파라미터와 변환 데이터값 검증 특정 상황에 대한 시스템 상태 검증
INFO	일반적인 애플리케이션 수행 정보	“프로그램 정보를 수정합니다” “프로그램 정보 수정을 완료했습니다”
WARN	주의해야할 정보	“입력 파라미터가 없습니다” “테스트입니다 필요 없는 로그를 삭제하세요”
ERROR	에러발생한 사항에 대한 정보	“입력값이 중복되었습니다”
FATAL	시스템 운영에 치명적인 이상 현상에 대한 정보	“Out Of Memory”가 발생했습니다

### 6.3. User Log Processing Rules

```
package com.tdtd.tmtd;  
  
import org.springframework.stereotype.Controller;  
import lombok.extern.slf4j.Slf4j;  
  
@Controller  
@Slf4j  
public class BoardController {
```

```

@GetMapping("/boardList.do")

public String boardList(Model model, String page) {

    log.info("BoardController 게시물 전체 값을 저장하여 이동하는 boardList");

    log.info("BoardController 게시물 조회 전달된 page : {}", page);


    return "boardList";

}

```

1. Annotation을 사용하여 @Slf4j를 선언한다.
2. Controller의 경우, 해당 컨트롤러의 역할 및 파라미터를 info로 처리한다.
3. 각 상황에 맞는 로그 레벨은 선택하여 처리한다.

클래스	설명	사용 예시
Dao	클래스명 + 메소드명 + Dao Access	log.info("UserDaoImpl selectListUser Dao Access");
Service	클래스명 + 메소드명 + Service 기능설명	log.info("UserServiceImpl selectListUser Service 기능설명");
Controller	클래스명 + 요청주소 + 메소드(GET,POST...) + 메시지 : parameter	log.info("UserController user.do GET 유저 상세정보 페이지 요청아이디 : {}", id);

## 7. Test Case Creation Rules

1. Junit4와 `sprint-test`를 사용하여 테스트 케이스를 작성한다.
2. `@Autowired`를 통하여 `Service`를 통합적으로 테스트한다.
3. Maven의 `test` 폴더인 `src/test/main`에 패키지를 작성하고 `JUnitTest4` 클래스를 생성한다.  
각 패키지는 `com.test.[branch name]`으로 하고 클래스명은 `[서비스]_JUnitTest`로 한다.
4. `@RunWith`, `@ContextConfiguration` Annotation 선언 및 `SpringJUnit4ClassRunner.class`, `ContextConfiguration(locations)`을 설정한다.
5. `@Transactional` Annotation 선언하여 테스트 후 `rollback`이 작동되도록 한다.

```
package com.test.sh;

import static org.junit.Assert.*;

import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.test.context.ContextConfiguration;
import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
import org.springframework.transaction.annotation.Transactional;

import com.tdtd.tmttd.model.service.ICommunityService;

@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations = {"file:src/main/webapp/WEB-INF/spring/**/*.xml"})
public class Test_JUnitTest {
```

**@Autowired**

**private ICommunityService service;**

**@Test**

**@Transactional**

public void test() {

fail("Not yet implemented");

}

}

## 8. Directory Structure

### 8.1. src/main/java

com.hcm.grw	Base Package
com.hcm.grw.ctrl	@Controller 파일
com.hcm.grw.aop	aop 파일
com.hcm.grw.comm	공통 모듈 파일
com.hcm.grw.config	API Configuration 파일
com.hcm.grw.filter	Filter 파일
com.hcm.grw.interceptor	Interceptor 파일
com.hcm.grw.model.mapper	@Repository 파일
com.hcm.grw.model.service	@Service 파일
com.hcm.grw.socket	WebSocket 파일
com.hcm.grw.dto	Dto/vo 파일

### 8.2. src/main/resources

root	Logback.xml 파일
config	myBatis Configuration xml 파일
properties	Oracle db properties 파일
sqls	Statement(Mapper) xml 파일

### 8.3. src/main/webapp

assets	bootstrap template
--------	--------------------

css	css 파일
font	font 파일
image	image 파일
js	JavaScript 파일
json	json 파일
libs	library 파일
WEB-INF	web.xml 파일
WEB-INF/error	error 페이지
WEB-INF/spring	spring bean configuration 파일
WEB-INF/views	jsp 파일

#### 8.4. src/test/java

com.test.(branch name)	JUnit Test 파일
------------------------	---------------