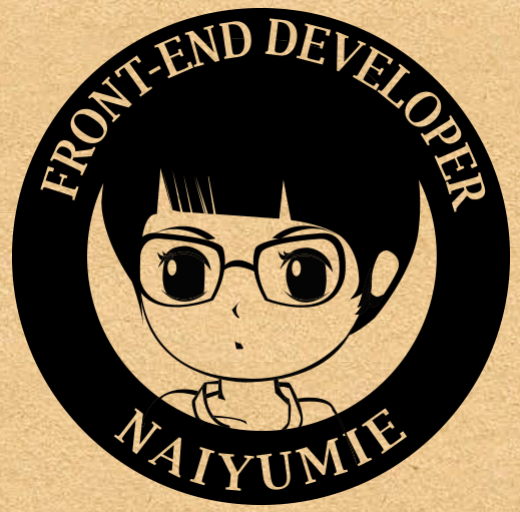


developer guide

Written by naiyumie

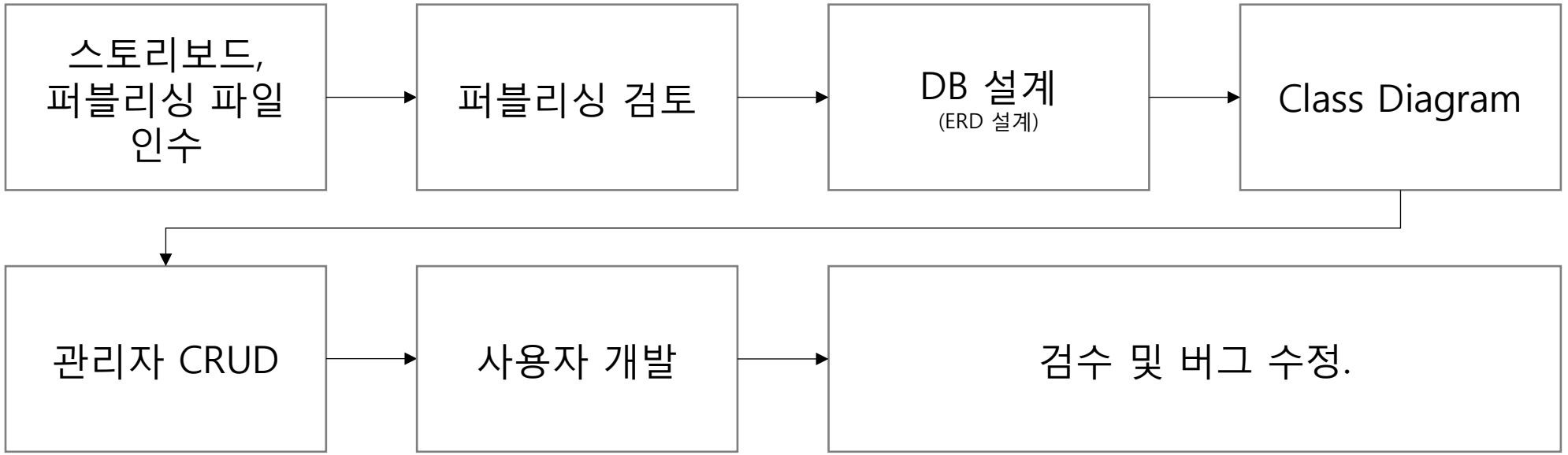
ver 1.0.0

개 정 이 력

No	버전	작성자	작성일자	변경사유	변경내용	승인자
1	1.0	원상필	2015.02.28	최초작성		

0 워킹 프로세스

웹개발 워킹 프로세스에 대해 설명 한다.



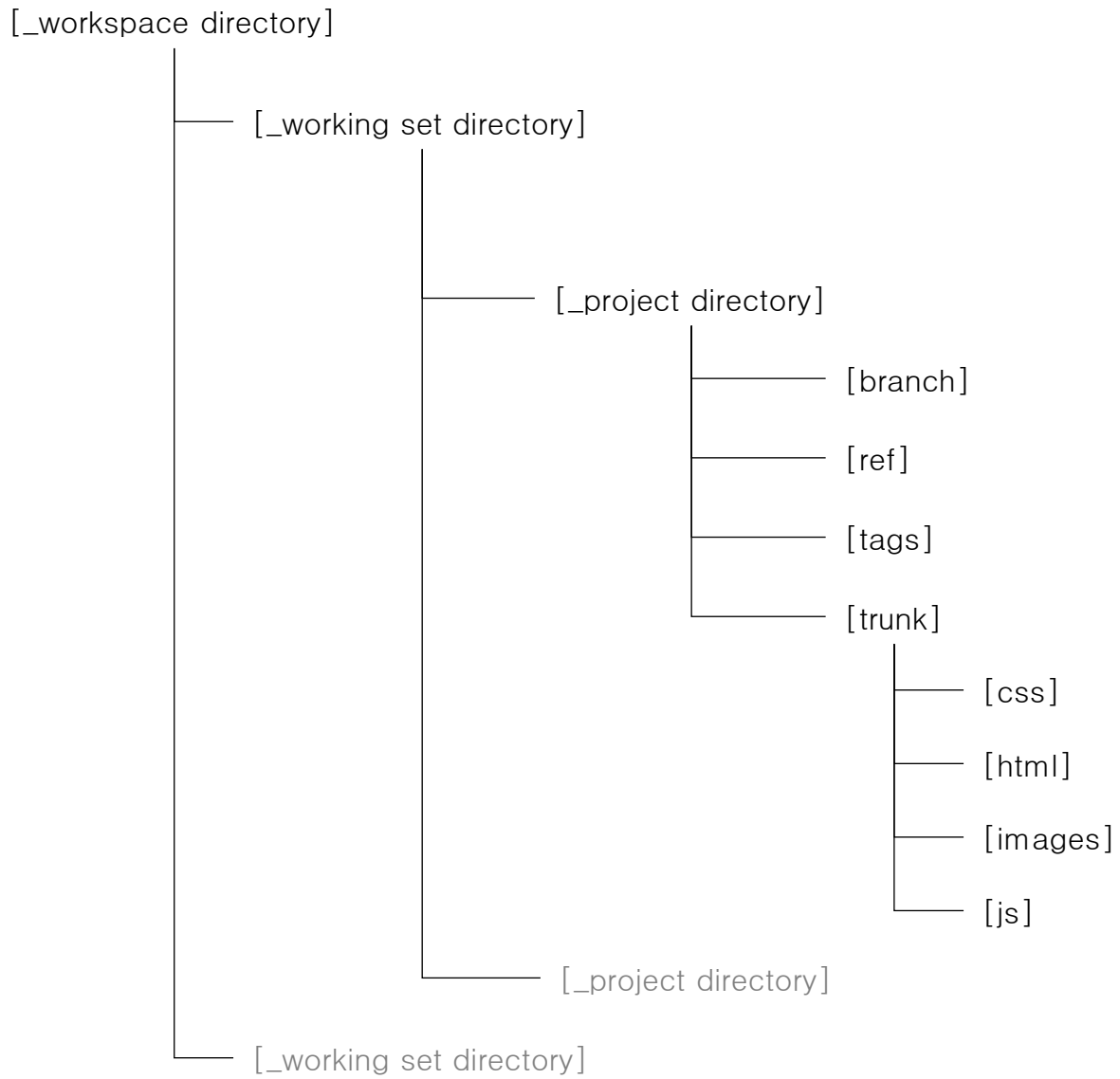
1. 스토리 보드와 퍼블리싱 작업을 인수 한다.
2. 퍼블리싱 파일을 검토 하여 필드의 누락, 자바스크립트 함수 호출 로직 파악을 한다.
3. DB설계 과정이다. ERD 설계를 한다.
4. Class Diagram을 그린다.
5. CRUD 멤버 함수를 작성하여 관리자를 개발한다.
    - 5.1 목록을 개발한다.
    - 5.1.1 목록에서 삭제 기능이 추가된다.
    - 5.1.2 검색 기능을 개발한다.
    - 5.2 조회(조회&업데이트)를 개발한다.
    - 5.3 업데이트를 처리한다.
    - 5.4 생성을 개발한다.
6. 사용자 부분에 표시할 내용을 개발한다.
    - 6.1 html을 검토하여 깨지지 않게 개발한다.
7. 검수 및 버그 수정을 진행 한다.

1. 디렉터리

디렉터리의 구조에 대해 설명 한다.

1.1.1 SVN을 기초한 디렉터리 구조#1

워크스페이스를 기초하여 워킹셋으로 구분하고, 각각의 프로젝트 디렉터리가 있으며, 각 프로젝트 디렉터리는 SVN을 기초한 디렉터리구조를 형상화 한다.



1.1.2 SVN을 기초한 디렉터리 구조#2

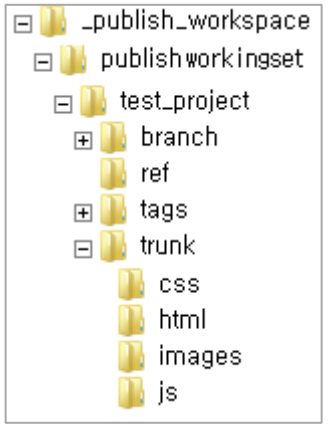
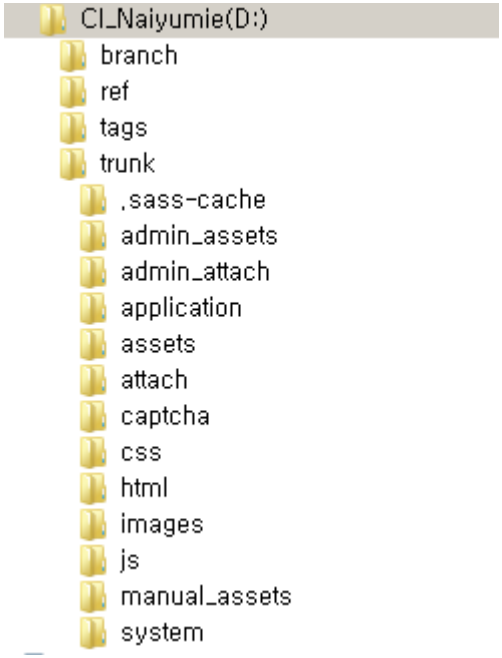


figure: 디렉터리 구조 예시



1.2 {projectDir}/ 디렉터리

프로젝트 루트 디렉터리는 기본적으로 branch, ref, tags, trunk의 4개의 디렉터리만 포함한다.

단, 개인 작업 스킬 역량에 따라 다양한 툴을 포함 할 수 있다.

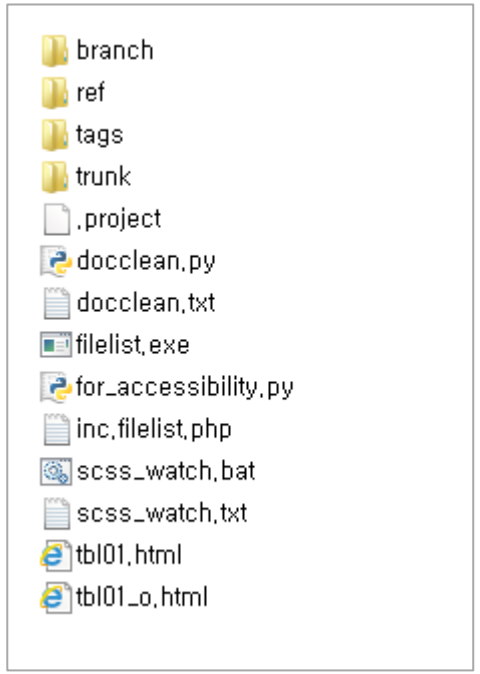


figure: 루트 디렉터리에 branch, ref, tags, trunk디렉터리를 포함하고
이클립스 연동시 필요한 .project파일
docclean.py filelist.exe등 다양한 툴을 포함하여 위치한 모습.

1.3.1 {projectDir}/branch/ 디렉터리

branch 디렉터리는 나무가지로서 나무줄기에서 뻗어나온 가지이다. 프로젝트 진행시 작은 프로젝트로 분류해서 개발하거나 따로 개발해야되는 경우에 사용한다. 프로젝트 안에 프로젝트 형태이다. 새로운 디렉터리와 소스코드가 들어간다.

이곳에 별도의 테스트사항이나 사용한 원본 라이브러리를 넣는다.

1.3.2 {projectDir}/ref/ 디렉터리

ref 디렉터리는 참조 문서가 위치한 곳으로써, 프로젝트 진행시 참조하여야 할 필수 문서 기타 부속 파일들을 위치한다.

참조 문서나 부속 파일을 넣는다. 개발에 사용된 다양한 문서 참조 프로그램을 넣는다.

GNB을 구현하기 위하여 이미지를 배열한 PSD를 남기거나, CI(로고)의 위치를 조절한 PSD를 갖고 있다면 이곳에 넣는다.

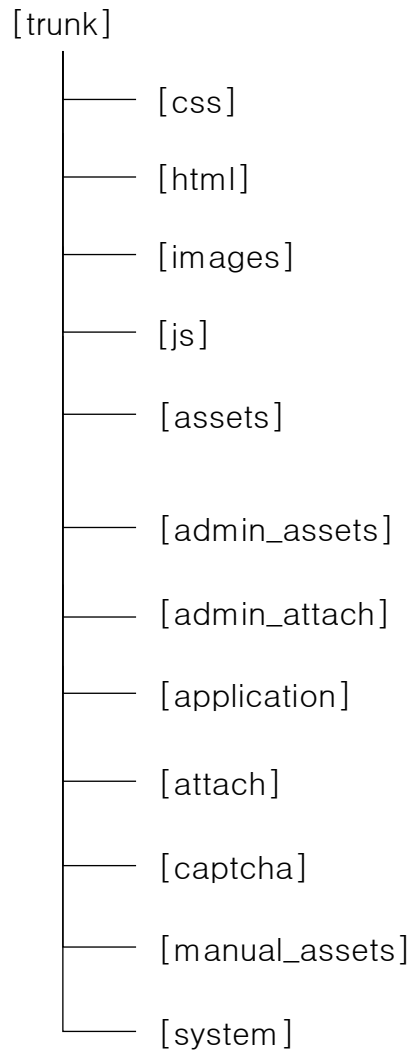
1.3.3 {projectDir}/tags/ 디렉터리

꼬리표라는 의미이다. 정기적으로 릴리즈되는 버번을 버전별로 디렉토리(꼬리표 형태)를 만들어 저장한다.
특히 tags 디렉토리 안에 버전명이 디렉토리 명으로 사용되어진다.

이곳에 타인에게 배포 하거나, 작업이 일시 중단 될 경우 압축하여 보관 한다.

1.3.4 {projectDir}/trunk/ 디렉터리

주 작업 디렉터리이다. 나무줄기, 몸통이라는 의미로 프로젝트에서 가장 중심이 되는 줄기이다. 모든 개발관련 작업은 이곳에서 수행된다.
이 줄기 아래에 프로젝트 디렉토리 구조 및 사용라이브러리, 코드가 들어간다.



이쪽 디렉터리들은 퍼블리싱 관련 파일로 ftrules 문서를 참조한다.

admin_assets 관리자 페이지는 퍼블리싱된 사용자 부분과 분리 되어있다. 이곳에 관리자용 클라이언트 파일이 위치 하고 있다.

admin_attach 관리자 페이지에서 프로그램에 의해 삽입되는 파일들이다.

attach 사용자 페이지에서 프로그램에 의해 삽입되는 파일 들이 위치한다.

captcha 캡챠가 위치한다.

manual_assets 매뉴얼 파일들이 위치한다.

application CI 구현 부분이다.

system CI의 코어 부분이다.

2. 컨벤션

파일명과 코딩 규칙에 대해 규약 한다.

CI 기본 규칙에 준순한다.

파일명은 영문~ 숫자로 구성 되어야 하며 dash(-)와 under-score(_)를 포함 할 수 있다.
영어를 제외한 한글/한자/일본어 등의 파일명은 인정하지 않는다.

tab&space는 4/4로 한다.

2.1 CodeIgniter 기본 확장사항 #1

Codeigniter의 General Style and Syntax을 기본적으로 따르나 아래의 예외가 있을 수 있다.
사고가 나거나 성능 하락이 있을 부분도 캣치하여 컨벤션에 집어 넣었다. 코드이그나이터 문서와 코드에서도 일관성이 없는 경우가 많다.

숏태그의 사용

서버 설정을 수정 할 수 없는 경우가 있을수 있으므로 숏태그를 사용 하지 않는다.

```
<??>틀림
<?=$a?>틀림
<?php ?> 맞음
<?php echo $a;?>맞음
```

File Format & Line Breaks

Line Ending - php(엔진이 아닌 file_get_content같은 함수)가 파싱 하지 않는 파일의 경우 CRLF도 큰 관계가 없다.

Code Indenting

```
function foo($bar) {     <- 틀림
}

foreach ($arr as $key => $val) { <- 맞음
}

if ($foo == $bar) { <- 맞음
} else {
}
```

제어문 반복문의 경우 이러한 문법도 크게 문제는 없으나 함수나 클래스는

```
function
{
}
```

와 같이 쓴다.

2.1.1 CodeIgniter 기본 확장사항 #2

PHP Closing Tag

/* End of file myfile.php */ /* Location: ./system/modules/mymodule/myfile.php */
이런식으로 경로를 표기하는데 /* eof */ /* End of file */로 표기해도 관계없다.

Class and File Names using Common Words

Pre같은 접두사를 쓰지 않아도 큰 관계는 없다.

Logical Operators

|| 오퍼레이터를 사용해도 큰 관계는 없다.

Commenting

개인적인 기호가 강한데 테스트 코드의 주석을 //로 표기하고 #을 주석으로 사용 하고 있다.
소스의 들여쓰기를 맞춰서 주석 처리한다.
이클립스로 주석 단축키로 주석을 지정하면 소스의 젤 앞부분에 주석 처리되어 블록을 깨트리므로
이클립스로 주석처리 할 경우 인덴션을 유지 한다.
PHPDoc은 고려 하지 않고 있다.

2.1.1 K&R Style에 기반한 코드이그나이터 코딩 컨벤션 스타일#1

1. 중괄호 쓰기
K&R 스타일은 제어문이나 반복문 중첩시 코드가 깔끔해 지고 쓸데 없이 코드가 많은 공간을 차지하지 않게 되므로 모양새가 좋아지는 점이 있다. 하지만 네임 스페이스. 클래스, 구조체, for, if 등 대부분의 경우 이런 스타일을 쓰지만
함수에 대해서는 시작 중괄호를 다음 줄에 쓴다.

```
// K&R Style
for (int i = 0; i < SOME_VALUE; i++) {
    if (SOME_CONDITION){
        SOME_STATMENT;
    } else {
        SOME_STATMENT;
    }
}

// K&R - if문
if (조건) {
  명령
}

// K&R - if 문에서 긴 조건들 ...
if ( THE_FIRST_CONDITION && NEXT_CONDITION &&
     ANOTHER_CONDITION && THE_LAST_CONDITION ) {
     SOME_STATMENT;
}

if (k == 5) {
     for (i = 0;i < 10;i++) {
          if (ar[i] == 10) {
               func(ar[i]);
          } else {
               proc(ar[i]);
          }
     }
}
```

2.1.2 K&R Style에 기반한 코드이그나이터 코딩 컨벤션 스타일#2

2. 클래스 함수 변수 상수 명 쓰기
K&R 스타일에서는 클래스, 함수에 대해서는 시작 중괄호를 다음 줄에 쓰나 Codeigniter기본 스타일에 따른다.

```
class Super_class {
        public function __construct()
        {
        }
}

// 모델
// 접미어로 m을 쓰고 있다.
class Model_examplem extends CI_Model {
}

// 컨트롤러
class Admin_example extends CI_Controller {
}

// 생성자 함수의 경우
public function __construct()
{
}
// 함수의 경우
// 함수는 문과()괄호를 붙인다.
public function index()
{
}

// 변수의 경우
// 배열의 경우 아래와 같이 쓴다.
$recent_joined_member
$model_data = array(
        'limit' => 10,
        'orderby' => 'dates'
);

//상수는 전부 대문자
COMPONET_INFO
```

2.2 controllers 작성 규칙

아래의 기본 컨트롤러 모양새를 유지한다. 주석의 형태는 달라져도 큰 관계가 없으나, 주석을 달지 않는 경우는 문제가 있다.
파일명 : application/controllers/admin_crud.php
관리자의 경우 admin 접두사가 붙는다.

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

/**************************************************************************************************

	Admin_crud
	기본적인 CRUD이다.

**************************************************************************************************/
class Admin_crud extends CI_Controller {
	/*********************************
		선언
	*********************************/
		# 전역 view_data
		var $view_data = array();
		# 컨텍스트
		var $context = 'admin_crud';

	/*********************************
		생성자
	*********************************/
		public function __construct()
		{
			parent::__construct();
		}

	/*********************************
		인덱스
	*********************************/
		public function index()
		{
			show_404();
		}
}
/* End of file */
```

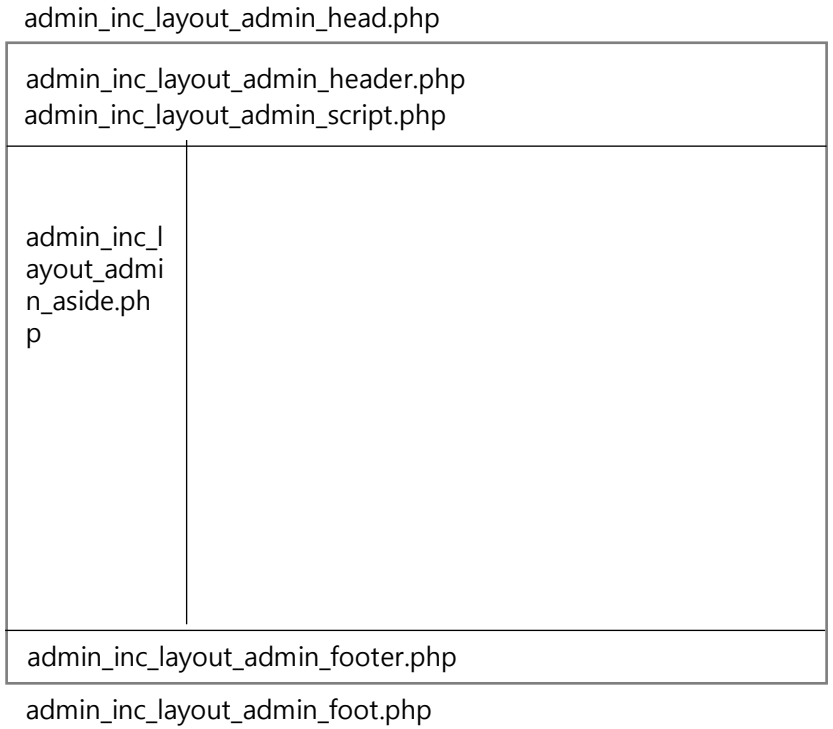
2.3 models 작성 규칙

아래의 기본 모델 모양새를 유지한다. 주석의 형태는 달라져도 큰 관계가 없으나, 주석을 달지 않는 경우는 문제가 있다.
파일명 : application/models/crudm.php
모델은 관리자여도 접두사가 붙지 않으며 접미사 m만 붙는다.

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

/**************************************************************************************************

        CRUD 모델

**************************************************************************************************/
class Crudm extends CI_Model {
	/*********************************
		선언
	*********************************/
		# 컨텍스트 - 테이블명
		var $context = 'crud';

	/*********************************
		생성자
	*********************************/
		function __construct()
		{
			parent::__construct();
		}

	/**************************************************************************************************

		CREATE

	**************************************************************************************************/

}
/* End of file */
```

2.4 views 작성 규칙

VIEW 파일의 컨벤션은 퍼블리싱에 기준하나, 관리자의 경우 부트스트랩을 쓰고 있다. 때문에 부트스트랩에 익숙해 질 필요가 있다.
admin_crud_create.php 생성의 경우 접미사로 _create가 붙는다.
admin_crud_list.php 목록의 경우 접미사로 _list가 붙는다.
admin_crud_retrieve_and_update.php 조회와 수정의 경우 _retrieve_and_update가 접미사로 붙는다.
만약 조회만 따로 작성 하는경우 admin_crud_retrieve.php
수정만 따로 작성하는 경우 admin_crud_update.php 와 같이 한다.



admin_inc_layout_admin_head.php
html헤드와 body오픈 사이의 태그가 위치한다.

admin_inc_layout_admin_header.php
gnb와 로고 사인인 정보를 표시하는 태그가 위치한다.

admin_inc_layout_admin_script.php
스크립트를 상단에 기술한다. head안에 있는게 정석이나 그 경우 상단 인클루드 처리가 복잡해진다. body 끝나는 태그 위에 있는 편이 성능상에 좋으나 스크립트 사용시 $(function(){}); 과 같이 호출 해주어야한다. 페이지 내에서 스크립트를 사용하지 않는다면 footer 밑쪽으로 내려도 관계 없다.

admin_inc_layout_admin_footer.php
관리자 카피라이터가 위치한다.

admin_inc_layout_admin_foot.php
body가 끝나는 태그가 위치하며 html끝나는 태그가 위치한다.

퍼블리싱(프론트엔드)과 달리 페이지내에 스크립트를 기술해도 큰관계는 없다.

```
<?php $this->load->view('admin_inc_layout_admin_footer');?>
<script type="text/javascript">
        // 네비게이션 활성화
        $("[data-sidebar_id=<?php echo $context?>]").addClass('active');
</script>
<?php $this->load->view('admin_inc_layout_admin_foot');?>
```

2.5 db 테이블 및 컬럼 작성 규칙

테이블과 컬럼명은 소문자, 숫자, 언더스코어만 허용되며 최대한 자연어로 기술한다.
연관성이 있다면 비슷하게 정렬되어 묶이도록 기술한다.

board
board_content
board_recommend

그러나 아래와 같이 쓰게되면 별도의 문서가 필요하게 된다.

BRD
BRD_C
BRD_R, B_TBL

enum타입의 경우 Y,N으로 기술한다. Y,N의 경우 별도의 아래 코드와 같이 처리하지 않아도 알아 볼 수 있기 때문이다.
true, false는 프로그래머만 아는 경우가 많다.

```
if ($text == 'Y') {
    return "예";
} else {
    return ="아니오";
}
```

dates, times는 컬럼을 구분한다. bootstrap에서 동시에 지원하지 않기 때문이고, 구분하는편이 검색에 유용하기 때문이다.
2000-01-01, 00:00:00로 초기화 한다. 실수로 값을 넣지 않아도 UI가 고장나지 않는다.
안타깝게도 Trigger는 MySql버전에 영향을 받아 지양하였다.

2.6 libraries 작성 규칙

아래의 기본 라이브러리 모양새를 유지한다. 주석의 형태는 달라져도 큰 관계가 없으나, 주석을 달지 않는 경우는 문제가 있다.
파일명 : application/libraries/Auth.php
라이브러리의 경우 CI가 정한 관례에 의해 파일명의 첫글자를 대문자로 한다.

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

/**************************************************************************************************

	Auth
	권한및 세션관련 라이브러리 이다.

**************************************************************************************************/
class Auth {
	/*********************************
		선언
	*********************************/
		var $ci;

	/*********************************
		생성자
	*********************************/
		public function __construct()
		{
			$this->ci =& get_instance();
		}

}

/* End of file */
```

2.7 helpers 작성 규칙

아래의 기본 헬퍼 모양새를 유지한다. 주석의 형태는 달라져도 큰 관계가 없으나, 주석을 달지 않는 경우는 문제가 있다.
파일명 : application/helpers/example_helper.php 헬퍼의 경우 CI가 정한 규칙에 의해 _helper 접미사가 붙는다.

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

# POST배열을 hidden으로 돌려준다.
function test()
{
	$ret = '';
	return $ret;
}
?>
```

3. CRUD 샘플 소개

CRUD샘플은 기본적인 데이터베이스 핸들링인 목록, 조회, 수정, 삭제를 관리자에 구현해놓은 샘플이다.
그러므로 CRUD샘플을 분석하면 코드의 흐름과 구현 스타일을 알 수 있다.
또한 새로운 CRUD모듈을 제작시 이 코드를 재활용하면 소중한 시간을 절약할 수 있다.

컨트롤러
application/controllers/admin_crud.php
라이브러리와 헬퍼의 호출, 모델의 호출, 페이지네이션, 뷰 파일들을 호출, 폼밸리데이션, 기본적인 폼 전송후 모델 처리를 배울수 있다.

모델
application/models/crudm.php
쓰기, 목록과 검색, 조회, 카운팅, 수정, 삭제처리를 배울수 있다.

뷰 - 생성
application/views/admin_crud_create.php
헤드,헤더,스크립트,풋,푸터를 view사이에 끼워넣어 레이아웃을 구성하고
인풋,텍스트에어리어,체크박스,레디오,셀렉트 태그와 밸리데이션 결과 값을 출력하는법,
그리고 폼전송을 배울 수 있다.

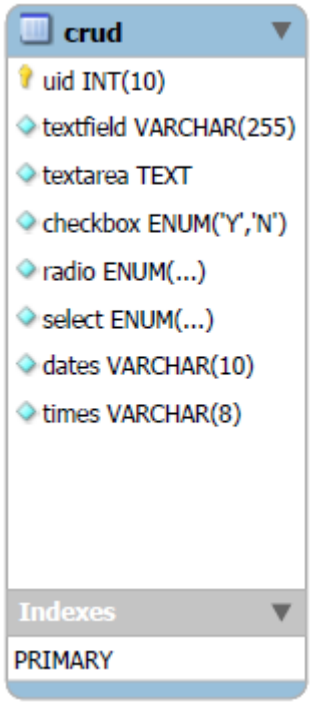
뷰 - 목록
application/views/admin_crud_list.php
데이터 리스팅과 검색을 배울수 있습니다. 페이지네이션을 사용 하는법을 배울수 있다.
삭제 처리를 하는 법도 배울수 있다.

뷰 - 조회 & 수정
application/views/admin_crud_retrieve_and_update.php
해당 컬럼을 조회하고 수정하는 법을 배울수 있다.

부트스트랩을 간략히 익힐수 있고 달력 UI, 시간 UI를 활용할 수 있게 된다.
그리고 CRUD샘플을 모두 공부하면 추가로 모듈을 개발할 준비를 마친것이다.

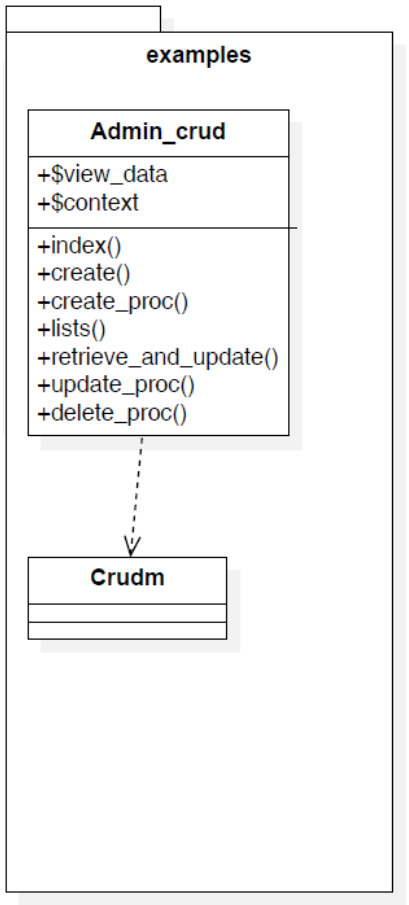
3.1 CRUD 샘플의 DB 테이블 설계 구조(erd)

테이블은 하나로 구성되어 있다.
기본적인 폼의 데이터 인풋과 DB와 연관성이 깊으므로 간략한 데이터 정규화를 배울수 있다.



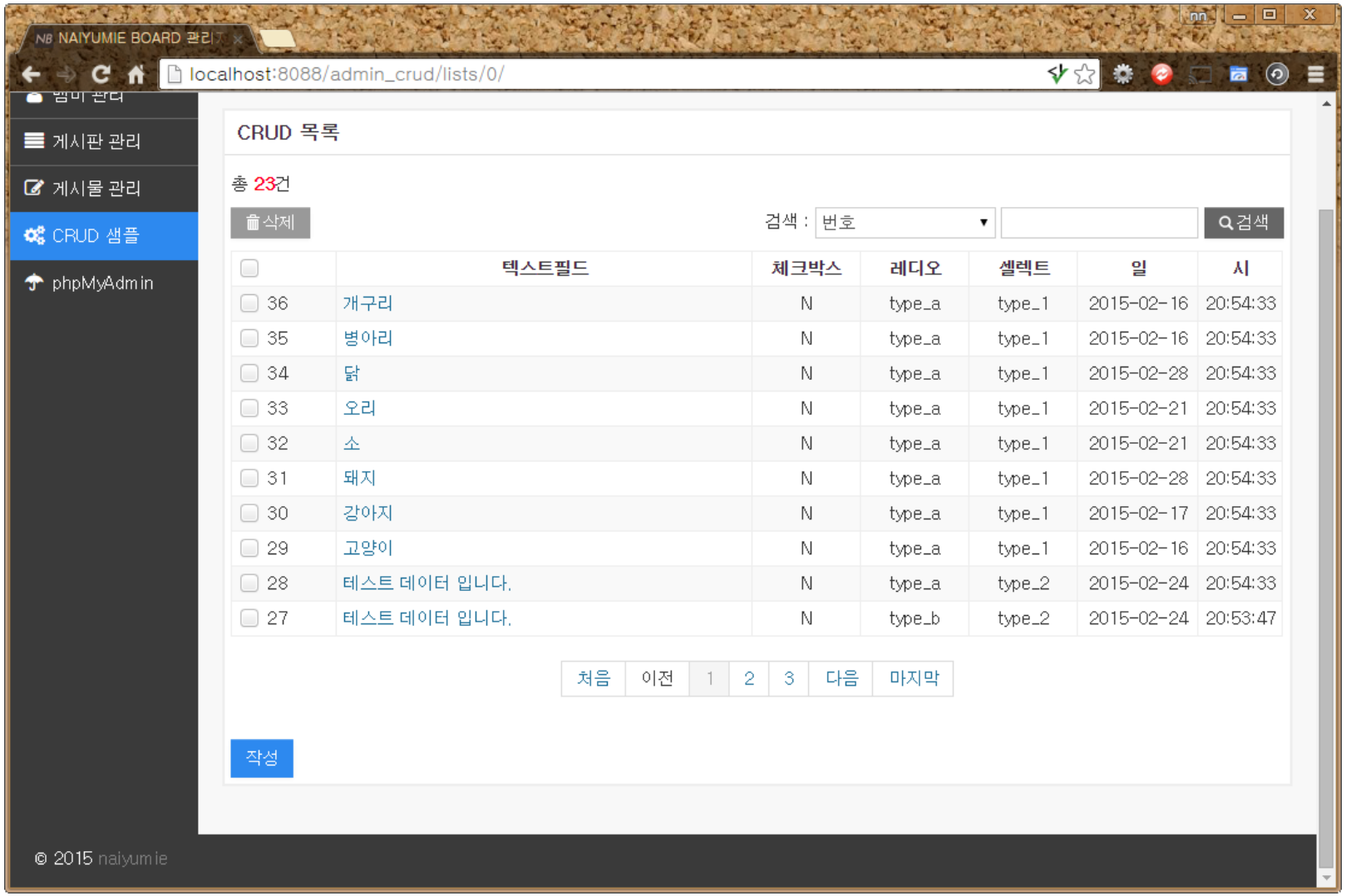
3.2 CRUD 샘플의 Class Diagram

클래스 다이어그램을 보고 컨트롤러와 모델이 어떤 관계에 있는지를 리뷰 한다.



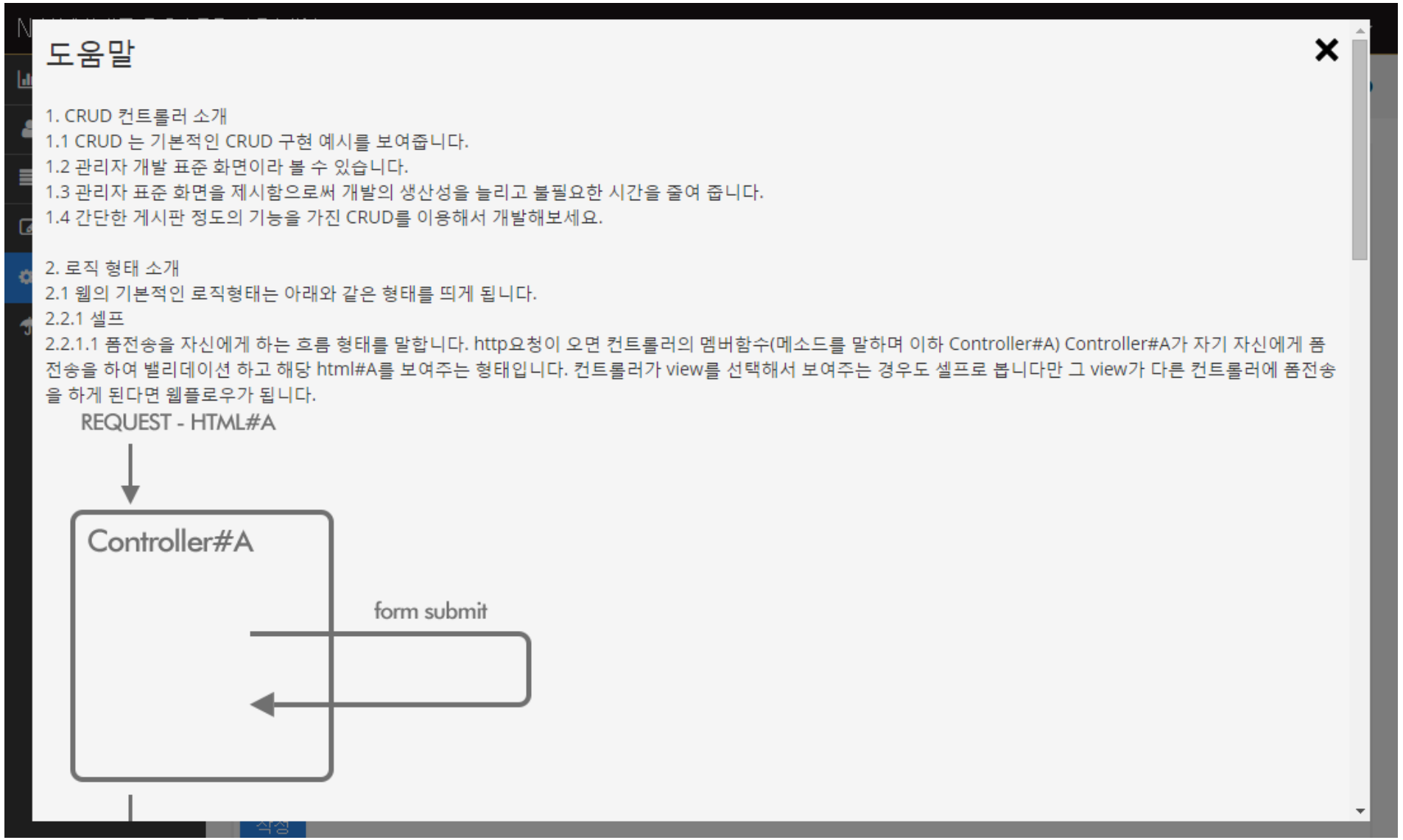
3.3 CRUD 샘플의 화면

화면을 보면 간략하게 어떠한 형태를 취하고 있는지 알 수 있다.



3.3 CRUD 샘플의 도움말

CRUD 샘플에서 (?) 물음표 마크를 클릭하면 여러 개발에대한 조언을 얻을 수 있다.



4. 기타

erd.mwb 파일은 mysql workbench 프로그램에서 사용하는 파일로 erd를 보여주고 DB를 변경해준다.

ClassDiagram.mdj 파일은 스타 UML 프로그램에서 사용하는 파일로 class diagram을 보여준다.