

Web Programming Laboratory

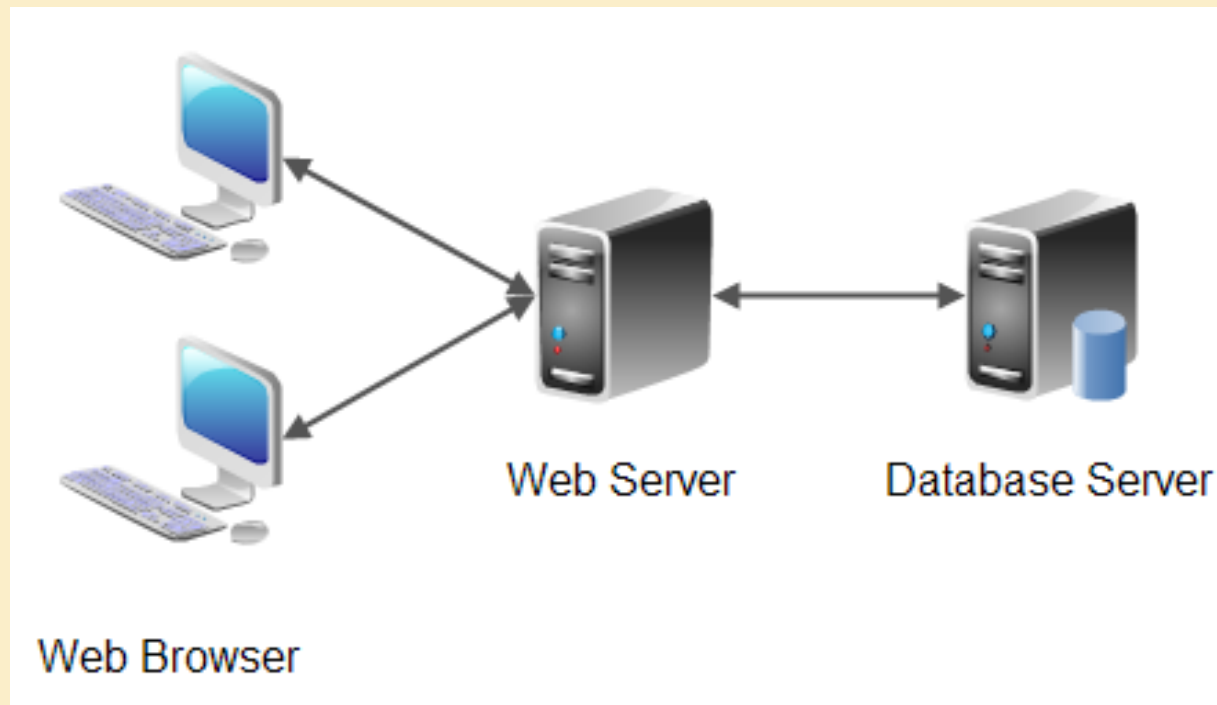
CSE-3200

Kazi Masudul Alam

Course Objective

- Learn web programming
- Build working web application in Groups
- Use home built SimTire v0.2 framework.

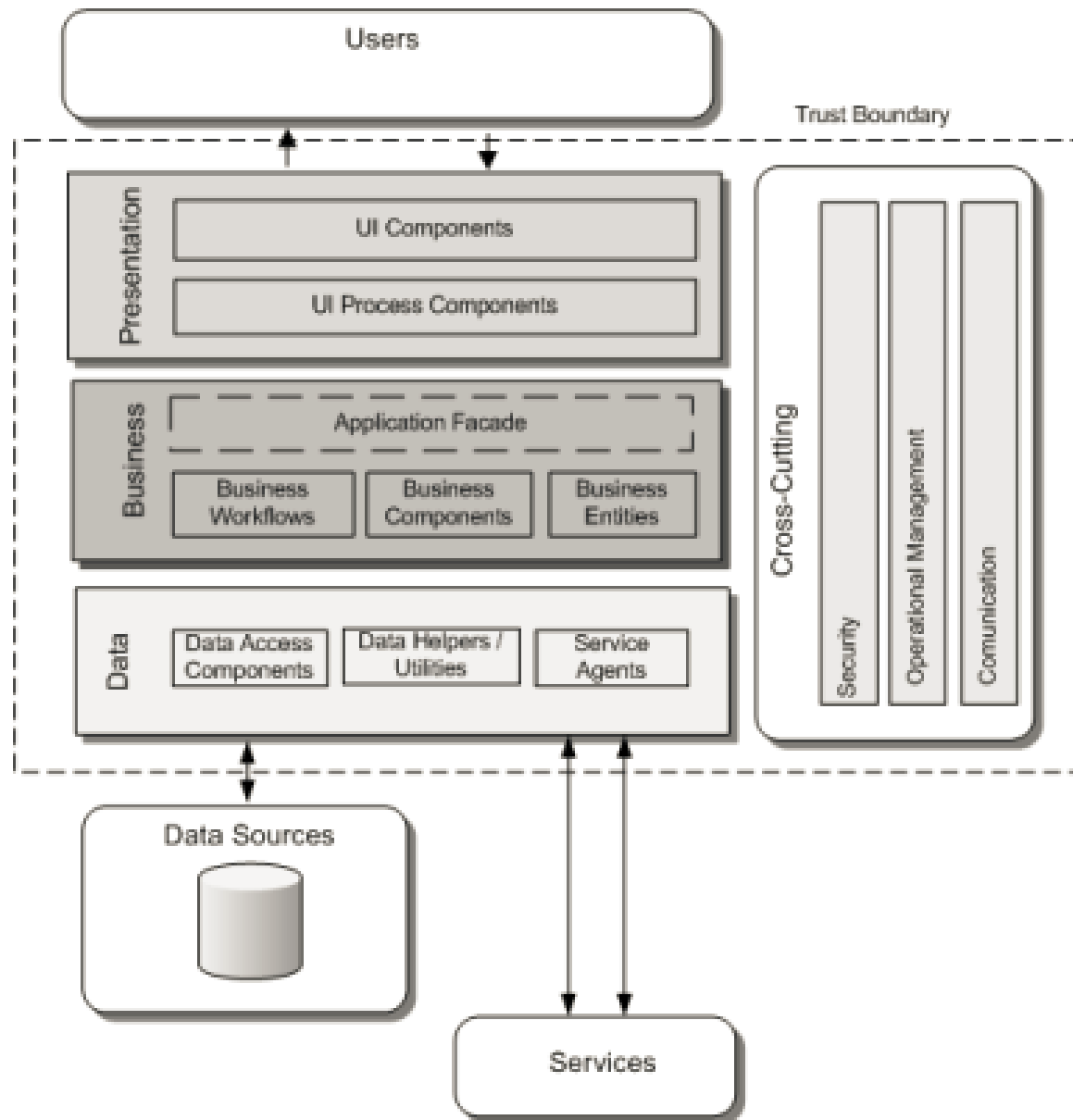
Web Application Architecture



Web Application Architecture

- Web Browser (Front End)
 - Html, CSS, JavaScript
- Web Server
 - Apache
 - Php (Backend)
- Database Server
 - MySQL

Layered Architecture



Project

- Modular University Automation
 - Build a new module or
 - Improve an existing module
- Each module will be built by one group
 - Each module and allowed group members are assigned in google project sheet.

Tools

- Software Development
 - Php, MySQL, Php IDE, WAMP server, Bootstrap CSS, JavaScript
- Documentation
 - Inline Comments in Code (**Vivid**)
- CVS
 - Bitbucket

Education Platform

			No of People	Project Code	
1	Improved CSE Discipline website (OOP)	filled	5	cseku_17_web	?
2	Improved advanced event calendar		2	cseku_17_ecal	
3	Improved news module		2	cseku_17_news	
4	Course Registration	KMA	0		
5	SimTire v0.2 Framework	KMA	0		
6	Marks Entry and Result Preparation	KMA	0		
7	Hall Management		2	cseku_17_hall	
8	Meal Management		2	cseku_17_meal	
9	Improved file-folder management system		3	cseku_17_file	
10	Library Management		3	cseku_17_library	
11	Medical Service Management		2	cseku_17_medical	
12	Improved Online Routine Preparation		3	cseku_17_routine	
13	Improved Project Archive		3	cseku_17_project	
14	Improved Survey System		2	cseku_17_survey	
15	Payment Management System		2	cseku_17_payment	
16	Improved Admission Test System		2	cseku_17_admission	
16	Transport Management System		2	cseku_17_transport	

Roles

- Assign each person a role in the project.
 - Team Leader (TL)
 - Developer (D)
 - Tester (T)

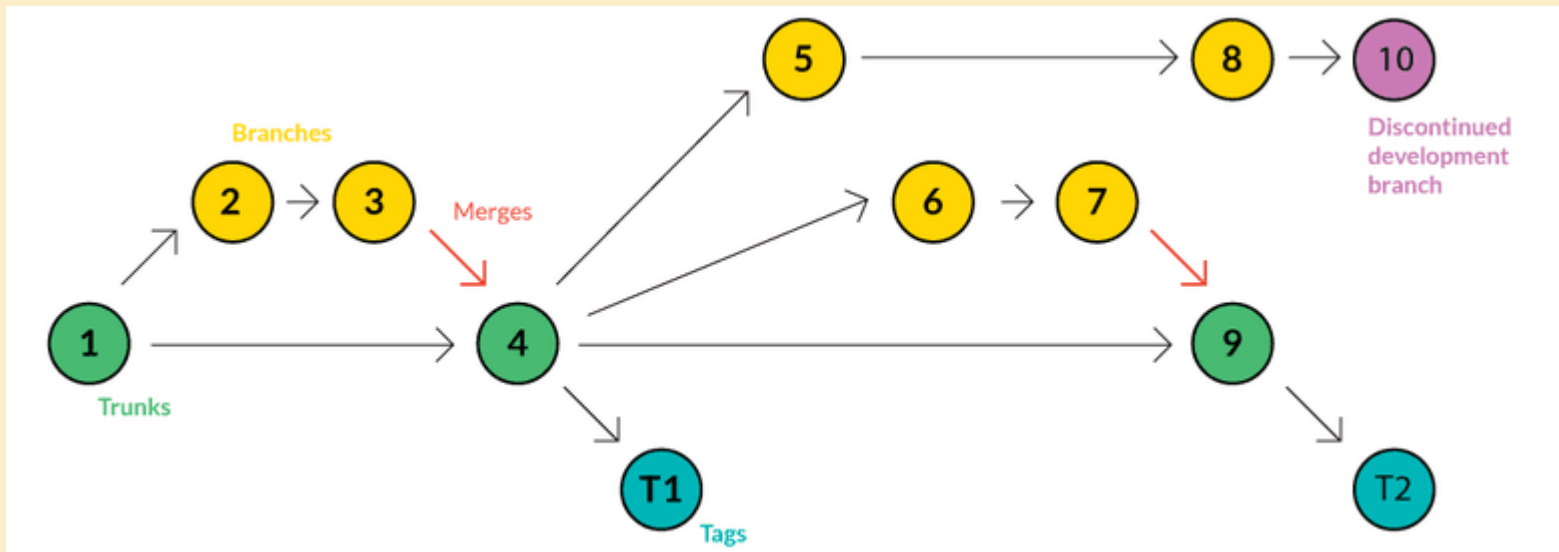
Student ID	Name	Email	Group Number	Role	Project Code
			1		
			1	TL	
			1		

Code Versioning System

Version Control

- Revision control is a kind of practice that **tracks and provides control over changes to source code**.
- Software developers use revision control software to **maintain documentation of change of source code**.
- It is common for **multiple versions** of the same software to be deployed in different sites and for the software's developers to be working simultaneously on updates.
- Bugs or features of the software are often only present in certain versions.
- It may be necessary to develop two versions of the software concurrently.
 - One version has bugs fixed, but no new features (**branch**), while the other version is where new features are worked on (**trunk**).

Version Control System	Repository Location	Checkout, Merge, Commit
SVN	Centralized	merge-before-commit
Git	Decentralized (or Distributed)	commit-before-merge
Mercurial	Decentralized (or Distributed)	commit-before-merge
Bazaar	Decentralized (or Distributed)	merge-before-commit



Bitbucket

- Bitbucket.org
- <https://bitbucket.org/kmalam/simtire>

The screenshot shows the Bitbucket web interface for a repository named 'simtire' owned by 'Kazi Masudul Alam'. The browser address bar shows the URL 'https://bitbucket.org/kmalam/simtire'. The left sidebar contains navigation links: Overview (selected), Source, Commits, Branches, Pull requests, Pipelines, Issues, Downloads, and Settings. Below these are options to 'SHARE YOUR THOUGHTS' by giving feedback or turning off new navigation. The main content area displays the repository overview, including the repository name, owner, and a table of statistics: 0 Open PRs, 1 Watcher, 1 Branch, and 0 Forks. A 'Send invitation' button is visible. Below the statistics is a section for 'Recent activity' showing three commits: one pushed to 'kmalam/based' by '9e166a5' 19 hours ago, another pushed to 'kmalam/based' by 'f1a2610' 2 days ago, and the repository creation by 'kmalam/based' 2 days ago. At the bottom of the main content area, there is a 'README' section for 'SimTire v0.2', which describes it as a simplified N-Tire framework for PHP web applications, currently in a beta stage. The author is identified as Dr. KAZI MASUDUL ALAM, Assistant Professor at CSE Discipline, Khulna University, Bangladesh.

Software Architecture Pattern

Architecture Pattern

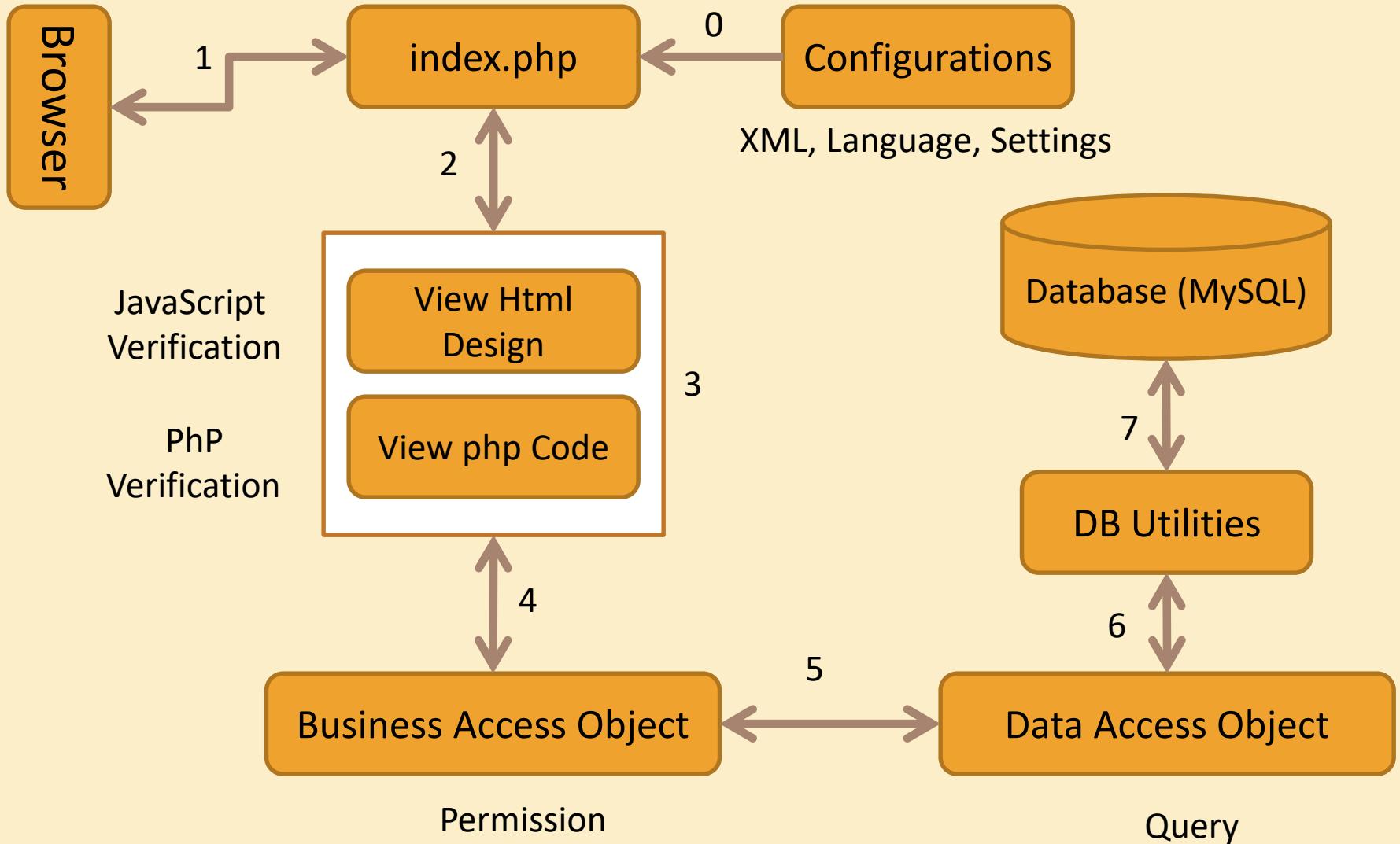
- Model-View-Controller Architecture
- 3-Tier Architecture
- Service Oriented Architecture

SimTier v0.2 (beta)

SimTier v0.2 (beta)

- A 3-tier framework for Php based web applications.
- An ongoing project.
- We will use this framework.


Framework Architecture



Framework Notes

- Folder structure
- Database
- Configuration
 - Permission
 - Navigation
- Page registration
- Login page
- Page Structure
- Business Flow
- XML and DB Utilities

Folder Structure

 common

 config


 docs

 lang

 logs

 modules


 resources


 template

 util


 .htaccess

 httpd-vhosts.conf.txt

 index.php

 index2.php

 install.php

 path.php

 readme.txt

Models and Other Settings

Navigation, Permission, Properties

Examples

Language file

LOG File

All the modules

Image, CSS, JavaScript

Look and Feel structure files

Various utility files

Web request mapping

All request comes here

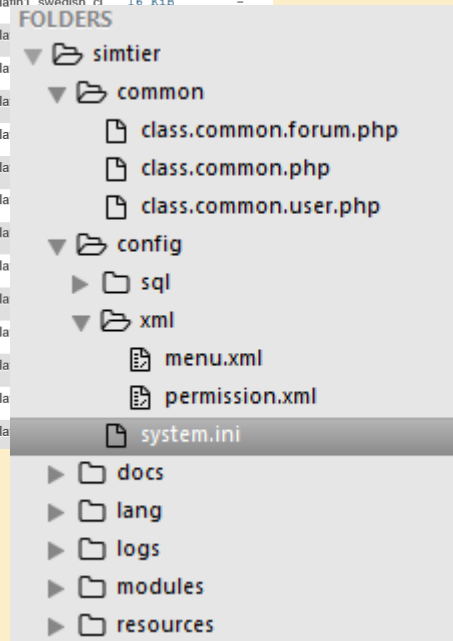
Installs permission

Path structure

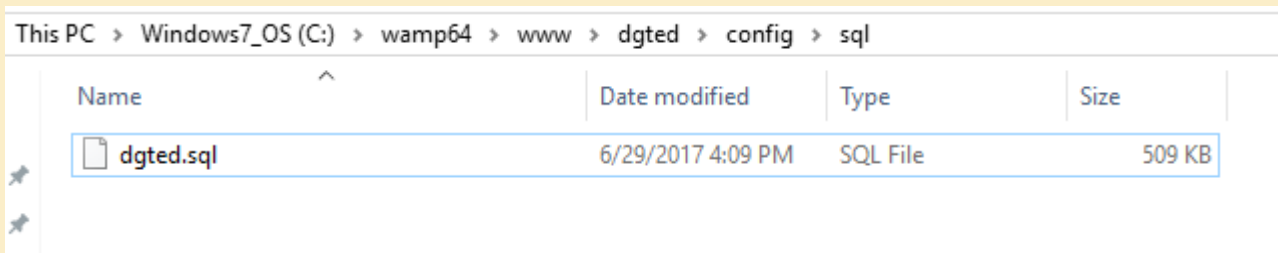
Database

Server: Local Databases > Database: based

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> tbl_discipline	★ Browse Structure Search Insert Empty Drop	7	InnoDB	latin1_swedish_ci	1.6 KiB	-
<input type="checkbox"/> tbl_discussion	★ Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	1.6 KiB	-
<input type="checkbox"/> tbl_discussion_category	★ Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	1.6 KiB	-
<input type="checkbox"/> tbl_discussion_comment	★ Browse Structure Search Insert Empty Drop	4	InnoDB	la		
<input type="checkbox"/> tbl_permission	★ Browse Structure Search Insert Empty Drop	44	InnoDB	la		
<input type="checkbox"/> tbl_position	★ Browse Structure Search Insert Empty Drop	13	InnoDB	la		
<input type="checkbox"/> tbl_role	★ Browse Structure Search Insert Empty Drop	6	InnoDB	la		
<input type="checkbox"/> tbl_role_permission	★ Browse Structure Search Insert Empty Drop	44	InnoDB	la		
<input type="checkbox"/> tbl_school	★ Browse Structure Search Insert Empty Drop	5	InnoDB	la		
<input type="checkbox"/> tbl_term	★ Browse Structure Search Insert Empty Drop	3	InnoDB	la		
<input type="checkbox"/> tbl_user	★ Browse Structure Search Insert Empty Drop	1	InnoDB	la		
<input type="checkbox"/> tbl_user_details	★ Browse Structure Search Insert Empty Drop	1	InnoDB	la		
<input type="checkbox"/> tbl_user_discipline	★ Browse Structure Search Insert Empty Drop	0	InnoDB	la		
<input type="checkbox"/> tbl_user_position	★ Browse Structure Search Insert Empty Drop	9	InnoDB	la		
<input type="checkbox"/> tbl_user_role	★ Browse Structure Search Insert Empty Drop	2	InnoDB	la		
<input type="checkbox"/> tbl_year	★ Browse Structure Search Insert Empty Drop	10	InnoDB	la		



```
system.ini
1 |; this is database config
2
3 db_host = "localhost"
4 db_username = "root"
5 db_password = ""
6 db_database = "based"
7
8
9 ; this is log config
10
11 log_enabled=true
12 ; info | debug | warning | error
13 log_level = "debug"
14 log_path = ".\logs\log.txt"
15
16 ; this is language config
17
18 lang_default="EN"
19
```



Configuration (Permission)

FOLDERS

- ▼ simtier
 - ▶ common
 - ▼ config
 - ▶ sql
 - ▼ xml
 - menu.xml
 - permission.xml
 - system.ini
 - ▶ docs
 - ▶ lang
 - ▶ logs
 - ▶ modules
 - ▶ resources
 - ▶ template
 - ▶ util
 - .htaccess
 - httpd-vhosts.conf.txt
 - index.php
 - index2.php
 - install.php
 - path.php
 - readme.txt

```
permission.xml x
79  <!-- DISCIPLINE Category -->
80  <permission>
81      <id>DISCIPLINE_C</id>
82      <name>DISCIPLINE_C</name>
83      <category>DISCIPLINE</category>
84  </permission>
85
86  <permission>
87      <id>DISCIPLINE_R</id>
88      <name>DISCIPLINE_R</name>
89      <category>DISCIPLINE</category>
90  </permission>
91
92  <permission>
93      <id>DISCIPLINE_U</id>
94      <name>DISCIPLINE_U</name>
95      <category>DISCIPLINE</category>
96  </permission>
97
98  <permission>
99      <id>DISCIPLINE_D</id>
100     <name>DISCIPLINE_D</name>
101     <category>DISCIPLINE</category>
102 </permission>
103
```

- Add your permissions for your module actions

Configuration (Navigation)

```
<!-- User -->
<Menu>
  <ParentMenu></ParentMenu>
  <Title>User</Title>
  <Permissions>
    <Permission>USER_R</Permission>
  </Permissions>
</Menu>

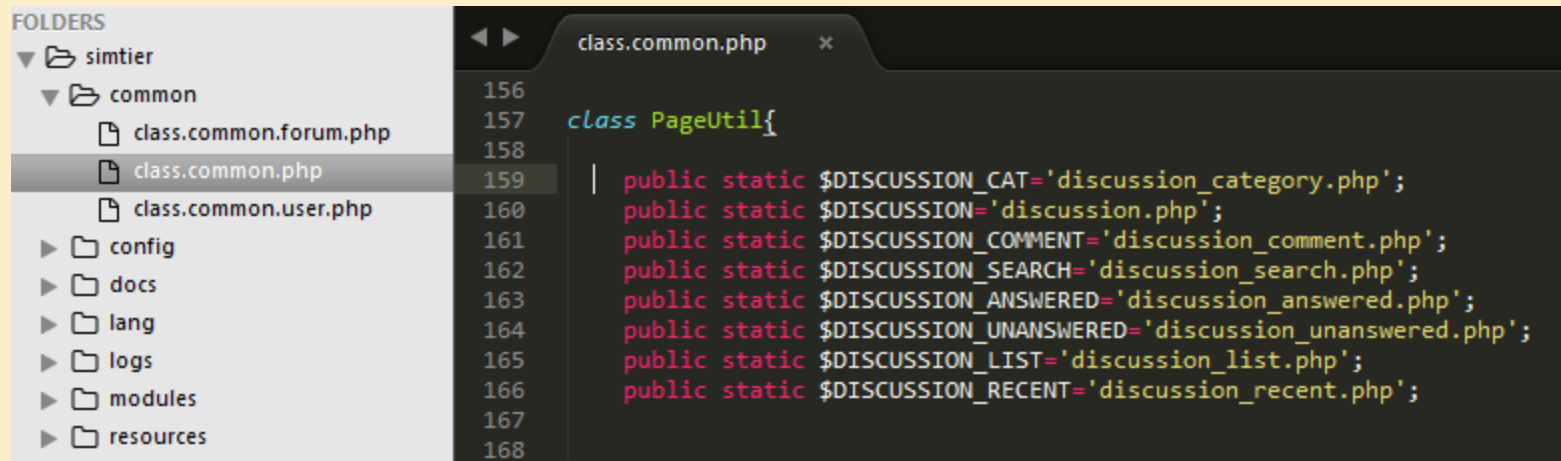
<Menu>
  <ParentMenu>User</ParentMenu>
  <Title>User Related</Title>
  <Permissions>
    <Permission>USER_R</Permission>
  </Permissions>
</Menu>

  <Menu>
    <ParentMenu>User Related</ParentMenu>
    <Title>User Details</Title>
    <Link>user_details.php</Link>
    <Permissions>
      <Permission>USER_R</Permission>
    </Permissions>
  </Menu>

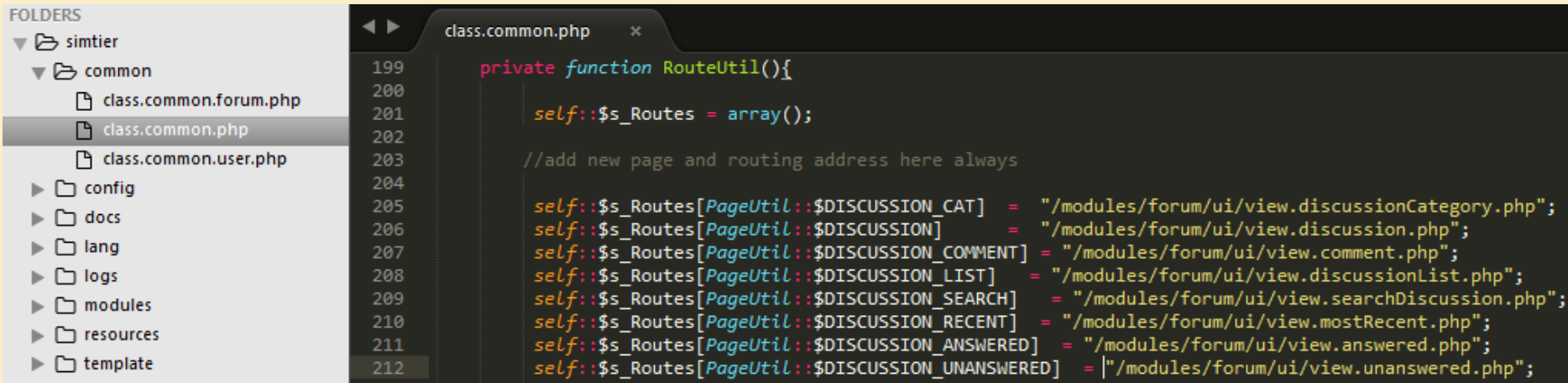
  <Menu>
    <ParentMenu>User Related</ParentMenu>
    <Title>Forgot Password</Title>
    <Link>forgot_password.php</Link>
    <Permissions>
      <Permission>USER_R</Permission>
    </Permissions>
  </Menu>
```

- Navigation is added in the menu.xml file
- Each menu is an xml item.
- Menu organization is highest two level.
- Each Menu can have a parent menu
 - Parent Menu
 - Menu
 - Menu

Page Registration



```
156
157 class PageUtil{
158
159     public static $DISCUSSION_CAT='discussion_category.php';
160     public static $DISCUSSION='discussion.php';
161     public static $DISCUSSION_COMMENT='discussion_comment.php';
162     public static $DISCUSSION_SEARCH='discussion_search.php';
163     public static $DISCUSSION_ANSWERED='discussion_answered.php';
164     public static $DISCUSSION_UNANSWERED='discussion_unanswered.php';
165     public static $DISCUSSION_LIST='discussion_list.php';
166     public static $DISCUSSION_RECENT='discussion_recent.php';
167
168
```



```
199     private function RouteUtil(){
200
201         self::$s_Routes = array();
202
203         //add new page and routing address here always
204
205         self::$s_Routes[PageUtil::$DISCUSSION_CAT] = "/modules/forum/ui/view.discussionCategory.php";
206         self::$s_Routes[PageUtil::$DISCUSSION] = "/modules/forum/ui/view.discussion.php";
207         self::$s_Routes[PageUtil::$DISCUSSION_COMMENT] = "/modules/forum/ui/view.comment.php";
208         self::$s_Routes[PageUtil::$DISCUSSION_LIST] = "/modules/forum/ui/view.discussionList.php";
209         self::$s_Routes[PageUtil::$DISCUSSION_SEARCH] = "/modules/forum/ui/view.searchDiscussion.php";
210         self::$s_Routes[PageUtil::$DISCUSSION_RECENT] = "/modules/forum/ui/view.mostRecent.php";
211         self::$s_Routes[PageUtil::$DISCUSSION_ANSWERED] = "/modules/forum/ui/view.answered.php";
212         self::$s_Routes[PageUtil::$DISCUSSION_UNANSWERED] = "/modules/forum/ui/view.unanswered.php";
```

- Add a new page in PageUtil
- Add view routes in the RouteUtil

Login Page

```
class.common.php
264 class MiddlewareUtil{
265
266     private static $s_Routes; //The single instance
267     private static $s_instance; //The single instance
268
269
270     private function MiddlewareUtil(){
271
272         self::$s_Routes = array();
273
274         //add which page should be successfully logged before getting to this page
275         //example: login.php should be successfully logged in to get to home.php
276         self::$s_Routes[PageUtil::$HOME] = PageUtil::$LOGIN ;
277
278
279         self::$s_Routes[PageUtil::$USER] = PageUtil::$LOGIN ;
280         self::$s_Routes[PageUtil::$USER_DETAILS] = PageUtil::$LOGIN ;
281
282         self::$s_Routes[PageUtil::$ROLE] = PageUtil::$LOGIN ;
283         self::$s_Routes[PageUtil::$PERMISSION] = PageUtil::$LOGIN ;
284
285
286         self::$s_Routes[PageUtil::$DISCIPLINE] = PageUtil::$LOGIN ;
287
288     }
289
```

- Add the login page as middleware before expected page

Page Structure

```
▼ modules
  ► dashboard
  ▼ forum
    ► bao
    ► dao
    ▼ ui
      ▼ blade
        view.answered.blade.php
        view.comment.blade.php
        view.discussion.blade.php
        view.discussionCategory.blade.php
        view.mostRecent.blade.php
        view.searchDiscussion.blade.php
        view.unanswered.blade.php
      view.answered.php
      view.comment.php
      view.discussion.php
      view.discussionCategory.php
      view.discussionList.php
      view.mostRecent.php
      view.searchDiscussion.php
      view.unanswered.php
```

```
view.discussion.php
1 |<?php
2
3 include_once 'blade/view.discussion.blade.php';
4 include_once '../common/class.common.php';
5 ?>
6
7
8 <div class="panel panel-default">
9
10 <div class="panel-heading">Discussion</div>
11
12 <div class="panel-body">
13
14 <div id="form">
15 <form method="post" class="form-horizontal">
16
17 <div class="form-group">
```

```
view.discussion.blade.php
1 |<?php
2
3 include_once '../util/class.util.php';
4 include_once '../../bao/class.discussionbao.php';
5 include_once '../../bao/class.discussionCategorybao.php';
6
7 $_DiscussionBAO = new DiscussionBAO();
8 $_DiscussionCategoryBAO = new DiscussionCategoryBAO();
9 $_DB = DBUtil::getInstance();
10
11 /* saving a new Discussion account*/
12 if(isset($_POST['save']))
13 {
14     $_DiscussionCategory = new DiscussionCategory();
15     $_Discussion = new Discussion();
16     $_Discussion->setID(Util::getGUID());
17     $_Discussion->setName($_DB->secureInput($_POST['txtQuestion']));
18     //$_Discussion->setCategory($_DB->secureInput($_POST['txtCat']));
19     $_Discussion->setTag($_DB->secureInput($_POST['txtTag']));
20     $_Discussion->setDescription($_DB->secureInput($_POST['txtDes']));
21 }
```

- Page is divided in two parts:
html view and php backend

Business Flow

```
view.discussion.blade.php x
1 |<?php
2
3 include_once './util/class.util.php';
4 include_once '../..../bao/class.discussionbao.php';
5 include_once '../..../bao/class.discussionCategorybao.php';
6
7 $_DiscussionBAO = new DiscussionBAO();
8 $_DiscussionCategoryBAO = new DiscussionCategoryBAO();
9 $_DB = DBUtil::getInstance();
10
11 /* saving a new Discussion account*/
12 if(isset($_POST['save']))
13 {
```



```
class.discussionbao.php x
1 |<?php
2
3 include_once './util/class.util.php';
4 include_once '../dao/class.discussiondao.php';
5
6
7 /*
8 Discussion Business Object
9 */
10 Class DiscussionBAO{
11
12 private $_DB;
13 private $_DiscussionDAO;
14
15 function DiscussionBAO(){
16
17     $this->_DiscussionDAO = new DiscussionDAO();
18
19 }
20
```



```
class.discussiondao.php x
1 |<?php
2 // write dao object for each class
3 include_once './common/class.common.php';
4 include_once './common/class.common.forum.php';
5 include_once './util/class.util.php';
6 include_once 'class.discussionCategorydao.php';
7
8 Class DiscussionDAO{
9
10 private $_DB;
11 private $_Cat;
12
13 function DiscussionDAO(){
14
15     $this->_DB = DBUtil::getInstance();
16     $this->_Cat = new DiscussionCategory();
17 }
18
19 // get all the Discussions from the database using the database query
20 public function getAllDiscussions(){
21
22     $DiscussionList = array();
23
24     $this->_DB->doQuery("SELECT * FROM tbl_discussion");
25
26     $rows = $this->_DB->getAllRows();
27
28     for($i = 0; $i < sizeof($rows); $i++) {
29         $row = $rows[$i];
30         $this->_Discussion = new Discussion();
31         $this->_Discussion->setID ( $row['ID']);
32
33     }
```

Utilities

FOLDERS

- ▼ simtier
 - ▶ common
 - ▼ config
 - ▶ sql
 - ▼ xml
 - menu.xml
 - permission.xml
 - system.ini
 - docs
 - lang
 - logs
 - modules
 - resources
 - template
 - ▼ util
 - class.util.php
 - class.xml.php
 - .htaccess
 - httpd-vhosts.conf.txt
 - index.php
 - index2.php
 - install.php
 - path.php
 - readme.txt

```
classutil.php x
71  /*
72  Get an instance of the Database
73  @return Instance
74  */
75  public static function getInstance() {
76      if(!self::$s_instance) { // If no instance then make one
77          self::$s_instance = new self();
78      }
79      return self::$s_instance;
80  }
81
82  // Magic method clone is empty to prevent duplication of connection
83  private function __clone() { }
84
85
86  /*
87  Reading database related configuration file
88  */
89  private function readConfig(){
90
91      $this->_ini_array = parse_ini_file("./config/system.ini");
92
93      $this->_host = $this->_ini_array['db_host'];
94      $this->_username = $this->_ini_array['db_username'];
95      $this->_password = $this->_ini_array['db_password'];
96      $this->_database = $this->_ini_array['db_database'];
97
98      $this->_log_enabled=$this->_ini_array['log_enabled'];
99      $this->_log_level=$this->_ini_array['log_level'];
100     $this->_log_file_path=$this->_ini_array['log_path'];
101
102
103     $this->_lang_default=$this->_ini_array['lang_default'];
104
105 }
106
```

Download

- <https://bitbucket.org/kmalam/simtier>

Documentation

How to Write Software Documentation

- Writing Software Documentation for Technical Users
 - Determine what information needs to be included.
 - Decide how much of the documentation should be within the program code and how much should be separate from it.
 - Choose the appropriate documentation tool.
- Writing Software Documentation for End Users
 - Determine the business reasons for your documentation.
 - Understand the audience you're writing the documentation for.
 - Determine the appropriate format(s) for the documentation.
 - Decide what form(s) the documentation should take.
 - Choose the appropriate documentation tool.

Thank You!