

UserPageKurtSys

Hi, I had some problems with graphviz. The problems seems to be that I wanted to use integer numbers (ID's). Changing it to VARCHAR-type (names instead of ID's), the problem was solved.

Below, the code of a first 'working' version of the project management. If someone has some ideas about features which should be included, please don't hesitate to let me know... Things start to work out fine now. Oh yeah, there might be a better way for date and time calculations and some MySQL-commands. Don't hesitate to let me know...

Very important right now: the project has to start with only 1 task and end with 1 task. Moreover, the starting time of that task has to be, for the start of the calculations, initially be set lower than all other starting times/dates. This behaviour may change in the future, once I decide how to do it the best way.

greetz,
Kurt.

MySQL databases

```
CREATE TABLE `tiki_projects` ( `projectID` int(14) NOT NULL auto_increment, `projectName` varchar(80) NOT NULL default '', `description` varchar(250) default NULL, `start` datetime NOT NULL default '0000-00-00 00:00:00', `finish` datetime NOT NULL default '0000-00-00 00:00:00', PRIMARY KEY (`projectID`), UNIQUE KEY `projectName` (`projectName`) ) TYPE=MyISAM AUTO_INCREMENT; CREATE TABLE `tiki_projects_tasks` ( `projectID` int(14) NOT NULL default '0', `taskID` int(14) NOT NULL auto_increment, `taskName` varchar(80) NOT NULL default '', `description` varchar(250) default NULL, `ES` datetime NOT NULL default '0000-00-00 00:00:00', `LS` datetime NOT NULL default '0000-00-00 00:00:00', `ECT` time default '00:00:00', `ECT_pes` time default NULL, `ECT_opt` time default NULL, `crash_ECT` time default NULL, `EF` datetime NOT NULL default '0000-00-00 00:00:00', `LF` datetime NOT NULL default '0000-00-00 00:00:00', `activityslack` time default '00:00:00', `est_cost` float default NULL, `crash_cost` float default NULL, PRIMARY KEY (`taskID`), UNIQUE KEY `taskName` (`taskName`), KEY `projectID` (`projectID`) ) TYPE=MyISAM AUTO_INCREMENT; CREATE TABLE `tiki_projects_tasks_predecessors` ( `taskID` int(14) NOT NULL default '0', `predecessorID` int(14) NOT NULL default '0', PRIMARY KEY (`taskID`, `predecessorID`) ) TYPE=MyISAM;
```

./lib/project/datetime.lib.php

```
<? function date_subdate($date1, $date2) { $diffseconds = strtotime($date1) - strtotime($date2); $days = floor($diffseconds/(24*3600)); $hours = floor(($diffseconds % 24*3600)/24); $minutes = floor(($diffseconds % 3600)/60); $seconds = $diffseconds % 60; return $days.' '.$hours.'.'.$minutes.'.'.$seconds; } function timestrtohour($time) { list($hour, $minute, $second) = split ("[:.-]", $time); $hour = $hour + $minute/60 + $second/3600; return $hour; } function hourtotimestr($hour) { $min = ($hour - floor($hour))*60; $sec = ($min - floor($min))*60; return floor($hour).'.'.$min.'.'.$sec; } ?>
```

./lib/project/project_graphlib.php

```
<?php class ProjectGraphLib { var $db; function ProjectGraphLib($db) { if (!$db) { die ("Invalid db object passed to ProjectGraphLib constructor"); } $this->db = $db; } function project_graph(&$str,
```

```

&$graph, $garg) { $task = $str['task']; $es = $str['ES']; $ls = $str['LS']; $ef = $str['EF']; $lf =
$str['LF']; $ect = $str['ECT']; $slack = $str['slack']; $graph->addAttributes(array( 'nodesep' =>
(isset($garg['att']['nodesep']))?$garg['att']['nodesep']:".7", 'rankdir' =>
(isset($garg['att']['rankdir']))?$garg['att']['rankdir']:'TB', 'size' =>
(isset($garg['att']['size']))?$garg['att']['size']:'6', 'bgcolor' =>
(isset($garg['att']['bgcolor']))?$garg['att']['bgcolor']:'transparent', 'URL' => 'tiki-project_tasks.php' ));
$graph->addNode("$task", array( // 'URL' => "tiki-index.php?page=" . urlencode(addslashes($page)),
'fontcolor' => (isset($garg['node']['fontcolor']))?$garg['node']['fontcolor']:'#007700', 'fontname' =>
(isset($garg['node']['fontname']))?$garg['node']['fontname']:"Arial", 'fontsize' =>
(isset($garg['node']['fontsize']))?$garg['node']['fontsize']:'9', 'label' =>
'<$task>'.(strlen($task)>29)?substr($task,0,28).'...:$task). //taskname '\n (ECT='substr($ect,
0,11)').' //estimated completion time '|{'substr($es, 0, 10).'|'.substr($ls, 0, 10).'|'. //early/late start
'|{'substr($slack, 0, 10).'|'. //activityslack '|{'substr($ef, 0, 10).'|'.substr($lf, 0, 10).'|'}', //early/late finish 'shape'
=> (isset($garg['node']['shape']))?$garg['node']['shape']:'Mrecord', 'color' => ($es ==
$ls)?(isset($garg['node']['cpcolor']))?$garg['node']['cpcolor']:'#CC0000':
(isset($garg['node']['color']))?$garg['node']['color']:'#007700', 'style' =>
(isset($garg['node']['style']))?$garg['node']['style']:'filled', 'fillcolor' =>
(isset($garg['node']['fillcolor']))?$garg['node']['fillcolor']:'#FFFFFF', 'width' =>
(isset($garg['node']['width']))?$garg['node']['width']:'0.8', 'height' =>
(isset($garg['node']['height']))?$garg['node']['height']:".1" ) ); //print("add node $task<br/>"); foreach
($str['next'] as $neig) { $this->project_graph($neig, $graph, $garg); $graph->addEdge( array("$task"
=> $neig['task']), array( 'style' => (isset($garg['edge']['style']))?$garg['edge']['style']:'solid', 'color' =>
(isset($garg['edge']['color']))?$garg['edge']['color']:'#007700' ) ); // print("add edge $task to
".$neig['task'].<br/>"); } } function get_graph_map($taskName, $garg) { $str =
$this->project_get_structure($taskName); $graph = new Image_GraphViz(); $this->project_graph($str,
$graph, $garg); return $graph->map(); } function project_get_structure($projectId, $predecessorID) {
if ($predecessorID=="") { $query = "select `taskID` from `tiki_projects_tasks` where
`projectId`='".$projectId.'" order by ES limit 1"; $result = $this->db->query($query); $result =
$result->fetchRow(); $predecessorID = $result['taskID']; } $query = "select `taskName`, `ES`, `LS`,
`ECT`, `EF`, `LF`, `activityslack` from `tiki_projects_tasks` where `taskID`=?"; $result =
$this->db->query($query, array($predecessorID)); $res = $result->fetchRow(); $aux['ES'] = $res['ES'];
$aux['LS'] = $res['LS']; $aux['ECT'] = $res['ECT']; $aux['EF'] = $res['EF']; $aux['LF'] = $res['LF'];
$aux['task'] = $res['taskName']; $aux['slack'] = $res['activityslack']; $query = "select `taskID` from
`tiki_projects_tasks_predecessors` where `predecessorID`='".$predecessorID.'" "; $result =
$this->db->query($query); $aux['next'] = array(); while ($res = $result->fetchRow()) { $aux['next'][] =
$this->project_get_structure($projectId, $res['taskID']); } return $aux; } } ?>

```

[./lib/project/project_taskslib.php](#)

```

<? class ProjectTasksLib { var $db; // PEAR db object to access database function ProjectTasksLib($db)
{ if (!$db) { die("Invalid db object passed to ProjectTasksLib constructor"); } $this->db = $db; }
function sql_error($query, $result) { trigger_error("MySQL error: ".$result->getMessage(). "in
query:<br/>".$query."<br/>", E_USER_WARNING); die; } function get_tasks($projectId) { $query =
"select * from `tiki_projects_tasks` where `projectId`=? order by `taskName`"; $result =
$this->db->query($query, array($projectId)); if (DB::isError($result)) { $this->sql_error($query,
$result); } $ret=Array(); while($res=$result->fetchRow(DB_FETCHMODE_ASSOC)) { $ret[]=$res; }
return $ret; } function get_taskdata($taskid) { $query = "select * from `tiki_projects_tasks` where
`taskID`=?"; $result = $this->db->query($query, array($taskid)); $ret =
$result->fetchRow(DB_FETCHMODE_ASSOC); $query = "select `predecessorID` from
`tiki_projects_tasks_predecessors` where `taskID`=?"; $result = $this->db->query($query,
array($ret['taskID'])); $ret['predecessors'] = array(); while

```

```

($res=$result->fetchRow(DB_FETCHMODE_ASSOC)) { $ret['predecessors'][] = $res['predecessorID'];
} return $ret; } function new_task($projectID, $taskname, $description, $sect, $sect_opt, $sect_pes,
$crash_ect, $es, $est_cost, $crash_cost) { $taskname = strip_tags($taskname); $description =
strip_tags($description); if ($es==0) { $es = date("YmdHis"); } $query = "insert into
`tiki_projects_tasks`(`projectID`, `taskName`, `description`, `ES`, `LS`, `ECT`, `ECT_opt`, `ECT_pes`,
`crash_ECT`, `est_cost`, `crash_cost`) values(?, ?, ?, ?, ?, ?, ?, ?, ?, ?)"; $result =
$this->db->query($query, array($projectID, $taskname, $description, $es, $es, $sect, $sect_opt,
$sect_pes, $crash_ect, $est_cost, $crash_cost)); if(DB::isError($result)) { $this->sql_error($query,
$result); } $query = "update `tiki_projects_tasks` set `EF`=date_add(`ES`, interval `ECT`
hour_second), `LF`=date_add(`ES`, interval `ECT` hour_second) where taskName=?"; $result =
$this->db->query($query, array($taskname)); if(DB::isError($result)) { $this->sql_error($query,
$result); } $query = "select `taskID` from `tiki_projects_tasks` where taskName=?"; $result =
$this->db->query($query, array($taskname)); if(DB::isError($result)) { $this->sql_error($query,
$result); } $res = $result->fetchRow(); return $res['taskID']; } function edit_task($taskid, $taskname,
$description, $sect, $sect_opt, $sect_pes, $crash_ect, $es, $est_cost, $crash_cost) { $description =
strip_tags($description); if ($es==0) { $es = date("YmdHis"); } $query = "update `tiki_projects_tasks`
set `taskName`=?, `description`=?, `ES`=?, `LS`=?, `ECT`=?, `ECT_opt`=?, `ECT_pes`=?,
`crash_ECT`=?, `est_cost`=?, `crash_cost`=? where `taskID`=?" ; $result = $this->db->query($query,
array($taskname, $description, $es, $es, $sect, $sect_opt, $sect_pes, $crash_ect, $est_cost, $crash_cost,
$taskid)); if(DB::isError($result)) { $this->sql_error($query, $result); } $query = "update
`tiki_projects_tasks` set `EF`=date_add(`ES`, interval `ECT` hour_second), `LF`=date_add(`ES`,
interval `ECT` hour_second) where taskID=?"; $result = $this->db->query($query, array($taskid));
return true; } function add_predecessor($taskid, $predid) { $query = "insert into
`tiki_projects_tasks_predecessors`(`taskID`, `predecessorID`) values(?, ?)"; $result =
$this->db->query($query, array($taskid, $predid)); if(DB::isError($result)) { $this->sql_error($query,
$result); } return true; } function remove_predecessors($taskid) { $query = "delete from
`tiki_projects_tasks_predecessors` where `taskID`=?" ; $result = $this->db->query($query,
array($taskid)); if(DB::isError($result)) { $this->sql_error($query, $result); } return true; } function
remove_task($taskid) { $query = "select `taskName` from `tiki_projects_tasks` where `taskID`=?" ;
$result = $this->db->query($query, array($taskid)); $result = $result->fetchRow(); $taskname =
$result["taskName"]; $query = "delete from `tiki_projects_tasks` where `taskID`=? limit 1"; $result =
$this->db->query($query, array($taskid)); $query = "delete from `tiki_projects_tasks_predecessors`
where `taskID`=? || `predecessorID`=?" ; $result = $this->db->query($query, array($taskid, $taskid));
if(DB::isError($result)) { $this->sql_error($query, $result); } return true; } function
calculate_chart_forward($projectID, $predecessorid, $pred_es, $pred_ect) { if ($predecessorid=="") {
$query = "select `taskID`, `ES`, `ECT` from `tiki_projects_tasks` where `projectID`=? order by ES asc
limit 1"; $result = $this->db->query($query, array($projectID)); if(DB::isError($result)) {
$this->sql_error($query, $result); } $result = $result->fetchRow(); $predecessorid = $result['taskID'];
$pred_es = $result['ES']; $pred_ect = $result['ECT']; $innerquery = "update `tiki_projects_tasks` set
`ES`=?, where projectID=?"; $result = $this->db->query($innerquery, array($pred_es, $projectID));
$innerquery = "update `tiki_projects_tasks` set `EF`=date_add(`ES`, interval `ECT` hour_second),
`LF`=date_add(`ES`, interval `ECT` hour_second) where `taskid`=?" ;
$result=$this->db->query($innerquery, array($predecessorid)); if(DB::isError($result)) {
$this->sql_error($query, $result); } } $query = "select `taskID` from
`tiki_projects_tasks_predecessors` where `predecessorID`=?" ; $result =
$this->db->query($query,array($predecessorid)); if(DB::isError($result)) { $this->sql_error($query,
$result); } while ($res = $result->fetchRow()) { $innerquery = "select `ES`, `ECT` from
`tiki_projects_tasks` where `taskID`=?" ; $result2 = $this->db->query($innerquery,
array($res['taskID'])); if(DB::isError($result)) { $this->sql_error($query, $result); } $res2 =
$result2->fetchRow(); // check if ES should be higher $innerquery = "select ? > date_add(?, interval ?
hour_second) as difference"; $innerresult = $this->db->query($innerquery, array($res2['ES'],
$pred_es, $pred_ect)); if(DB::isError($innerresult)) { $this->sql_error($query, $result); } $innerres =
$innerresult->fetchRow(); $difference = $innerres['difference']; if ($difference==0) { // update ES, EF

```

```

and LF $innerquery = "update `tiki_projects_tasks` set `ES`=date_add(?, interval ? hour_second)
where `taskID`=?"; $innerresult = $this->db->query($innerquery,array($pred_es, $pred_ect,
$res['taskID'])); if(DB::isError($innerresult)) { $this->sql_error($query, $result); } $innerquery =
"update `tiki_projects_tasks` set `LF`=date_add(`ES`, interval `ECT` hour_second),
`EF`=date_add(`ES`, interval `ECT` hour_second) where `taskID`=?"; $innerresult =
$this->db->query($innerquery, array($res['taskID'])); if(DB::isError($innerresult)) {
$this->sql_error($query, $result); } $innerquery = "select `ES` from `tiki_projects_tasks` where
`taskID` = ?"; $innerresult = $this->db->query($innerquery, array($res['taskID']));
if(DB::isError($result)) { $this->sql_error($query, $result); } $innerres = $innerresult->fetchRow();
$res2['ES'] = $innerres['ES']; } $this->calculate_chart_forward($projectID, $res['taskID'], $res2['ES'],
$res2['ECT']); } return true; } function calculate_chart_backward($projectID, $taskid, $task_lf,
$task_ect) { if ($taskid=="") { $query = "select `taskID`, `LF`, `ECT` from `tiki_projects_tasks` where
`projectID`=? order by EF desc limit 1"; $result = $this->db->query($query, array($projectID));
$result = $result->fetchRow(); $taskid = $result['taskID']; $task_lf = $result['LF']; $task_ect =
$result['ECT']; $query = "update `tiki_projects_tasks` set LF=?, activityslack=0 where projectID=?";
$result = $this->db->query($query, array($task_lf, $projectID)); $query = "update
`tiki_projects_tasks` set LS=date_sub(`LF`, interval `ECT` hour_second) where projectID=?";
$innerresult = $this->db->query($query, array($projectID)); } $query = "select `predecessorID` from
`tiki_projects_tasks_predecessors` where `taskID`=?"; $result = $this->db->query($query,
array($taskid)); while ($res = $result->fetchRow()) { $innerquery = "select `LF`, `ECT` from
`tiki_projects_tasks` where `taskID`=?"; $result2 = $this->db->query($innerquery,
array($res['predecessorID'])); if(DB::isError($result)) { $this->sql_error($query, $result); } $res2 =
$result2->fetchRow(); $innerquery = "select ? < date_sub(?, interval ? hour_second) as difference";
$innerresult = $this->db->query($innerquery, array($res2['LF'], $task_lf, $task_ect));
if(DB::isError($result)) { $this->sql_error($query, $result); } $innerres = $innerresult->fetchRow();
$difference = $innerres['difference']; if ($difference==0) { // update LF and slack $innerquery =
"update `tiki_projects_tasks` set `LF`=date_sub(?, interval ? hour_second) where `taskID`=?";
$this->db->query($innerquery, array($task_lf, $task_ect, $res['predecessorID'])); $innerquery =
"update `tiki_projects_tasks` set `LS`=date_sub(`LF`, interval `ECT` hour_second) where `taskID`=?";
$this->db->query($innerquery, array($res['predecessorID'])); $innerquery = "select `EF`, `LF` from
`tiki_projects_tasks` where `taskID`=?"; $innerresult = $this->db->query($innerquery,
array($res['predecessorID'])); if(DB::isError($innerresult)) { $this->sql_error($query, $innerresult); }
$innerres = $innerresult->fetchRow(); $slack=date_subdate($innerres['LF'], $innerres['EF']);
$innerquery = "update `tiki_projects_tasks` set `activityslack`=? where `taskID`=?";
$this->db->query($innerquery, array($slack, $res['predecessorID'])); $res2['LF'] = $innerres['LF']; }
$this->calculate_chart_backward($projectID, $res['predecessorID'], $res2['LF'], $res2['ECT']); } return
true; } } ?>

```

</lib/project/projectlib.php>

```

<? class ProjectsLib { var $db; // PEAR db object to access database function ProjectsLib($db) { if
(!$db) { die("Invalid db object passed to ProjectTasksLib constructor"); } $this->db = $db; } function
sql_error($query, $result) { trigger_error("MYSQL error: ".$result->getMessage())."in
query:<br/>".$query."<br/>",E_USER_WARNING); die; } function get_projects() { $query = "select *
from `tiki_projects` order by `projectName`"; $result = $this->db->query($query); if
(DB::isError($result)) { $this->sql_error($query, $result); } $ret=Array();
while($res=$result->fetchRow(DB_FETCHMODE_ASSOC)) { $ret[]=$res; } return $ret; } function
get_projectdata($projectID) { $query = "select * from `tiki_projects` where `projectID`=?"; $result =
$this->db->query($query, array($projectID)); $ret = $result->fetchRow(DB_FETCHMODE_ASSOC);
return $ret; } function new_project($projectname, $description, $start, $finish) { $projectname =
addslashes(strip_tags($projectname)); $description = addslashes(strip_tags($description)); if

```

```

($start==0) { $start = date("YmdHis"); } if ($finish==0) { $finish = date("YmdHis"); } $query =
"insert into `tiki_projects` (`projectName`, `description`, `start`, `finish`) values(?, ?, ?, ?)"; $result =
$this->db->query($query, array($projectname, $description, $start, $finish)); if(DB::isError($result)) {
$this->sql_error($query, $result); } return true; } function edit_project($projectname, $description,
$start, $finish) { $description = addslashes(strip_tags($description)); if ($start==0) { $start =
date("YmdHis"); } if ($finish==0) { $finish = date("YmdHis"); } $query = "update `tiki_projects` set
`description`=?, `start`=?, `finish`=? where `projectName`=?" ; $result = $this->db->query($query,
array($description, $start, $finish, $projectname)); if(DB::isError($result)) { $this->sql_error($query,
$result); } return true; } function remove_project($projectid) { $query = "delete from `tiki_projects`
where `projectID`=? limit 1"; $result = $this->db->query($query, array((int)$projectid));
if(DB::isError($result)) { $this->sql_error($query, $result); } $query = "select `taskName` from
`tiki_projects_tasks` where `projectID`=?" ; $result = $this->db->query($query, array($projectid));
while ($res=$result->fetchRow()) { $task = $res['taskName']; $query = "delete from
`tiki_projects_tasks_predecessors` where `taskName`=?" ; $res2 = $this->db->query($query,
array($task)); if(DB::isError($result)) { $this->sql_error($query, $result); } } $query = "delete from
`tiki_projects_tasks` where `projectID`=?" ; $result = $this->db->query($query, array($projectid));
if(DB::isError($result)) { $this->sql_error($query, $result); } return true; } } ?>

```

./templates/tiki-admin_project_tasks.tpl

```

<H1>Project (ID: {$projectID})</h1> <p><a href="tiki-admin_projects.php" class="tablink">Admin
projects</a></p> <H2>Add task to project</h2> <form method="post" action="tiki-
admin_project_tasks.php?projectID={$projectID}"> <table> <tr> <td>Taskname (unique):
</td><td> <input type="text" name="taskname" value="{ $taskdata.taskName}" size="30"
maxlength="160" /> (id: <input type="text" name="taskid" value="{ $taskdata.taskID}" readonly
size="10" maxlength="14" />) </td> </tr><tr> <td>Description: </td><td> <textarea
name="description" rows="5" cols="80">{$taskdata.description}</textarea> </td> </tr><tr>
<td>Earliest starting time: </td><td> <input type="text" name="es" value="{ $taskdata.ES}"
size="19" maxlength="19" /> </td> </tr><tr> <td>Completion time:</td> </tr><tr> <td>- most
likely </td><td> <input type="text" name="ect" value="{ $taskdata.ECT}" size="10" maxlength="10"
/> </td> </tr><tr> <td>- optimistic </td><td> <input type="text" name="ect_opt"
value="{ $taskdata.ECT_opt}" size="10" maxlength="10" /> </td> </tr><tr> <td>- pessimistic
</td><td> <input type="text" name="ect_pes" value="{ $taskdata.ECT_pes}" size="10"
maxlength="10" /> </td> </tr><tr> <td>- crash </td><td> <input type="text" name="crash_ect"
value="{ $taskdata.crash_ECT}" size="10" maxlength="10" /> </td> </tr><tr>
<td>Costs/hour:</td> </tr><tr> <td>- estimated </td><td> <input type="text" name="est_cost"
value="{ $taskdata.est_cost}" size="15" maxlength="20" /> </td> </tr><tr> <td>- crash </td><td>
<input type="text" name="crash_cost" value="{ $taskdata.crash_cost}" size="15" maxlength="20" />
</td> </tr><tr> <td>Possible predecessors: </td><td> <table class="normal" width="100%"> <tr>
<td class="heading">&nbsp;</td> <td class="heading">ID</td> <td class="heading">name</td>
<td class="heading">description</td> </tr> {section name=i loop=$tasks} {if $tasks[i].taskID !=
$taskdata.taskID} <tr> <td class="even"><input type="checkbox" name="predecessors[]"
value="{ $tasks[i].taskID|escape}" {if in_array($tasks[i].taskID, $taskdata.predecessors)}checked{/if}
/></td> <td class="even">{$tasks[i].taskID|escape}</td> <td
class="even">{$tasks[i].taskName|truncate:40:"..."}</td> <td
class="even">{$tasks[i].description|truncate:100:"..."}</td> </tr> {/if} {/section} </table> </td>
</tr> </table> <input type="submit" name="{ $send}" value="add/edit task" /> </form>
<H2>Project tasks</h2> <center> <p> <span class="tabbut"> <a href="tiki-
admin_project_tasks.php?projectID={$projectID}&calculate_cpchart=true"
class="tablink">Calculate</a> </span> <span class="tabbut"> <a href="tiki-
project_graph.php?projectID={$projectID}" target="_blank" class="tablink">Critical path chart</a>
</span> </p> </center> <table class="normal" width="100%"> <tr> <td class="heading">ID</td>

```

```

<td class="heading">name</td> <td class="heading">ES</td> <td class="heading">ECT</td> <td
class="heading">LF</td> <td class="heading">slack</td> <td class="heading">action</td> </tr>
{section name=t loop=$tasks} <tr> {if $smarty.section.t.index % 2} <td
class="odd">{$tasks[t].taskID}</td> <td class="odd">{$tasks[t].taskName}</td> <td
class="odd">{$tasks[t].ES}</td> <td class="odd">{$tasks[t].ECT}</td> <td
class="odd">{$tasks[t].LF}</td> <td class="odd">{$tasks[t].activityslack}</td> <td
class="odd"><a class="link" href="tiki-
admin_project_tasks.php?projectID={$projectID}&remove={$tasks[t].taskID}"></a> <a class="link" href="tiki-
admin_project_tasks.php?projectID={$projectID}&edit={$tasks[t].taskID}"></a></td> {else} <td
class="even">{$tasks[t].taskID}</td> <td class="even">{$tasks[t].taskName}</td> <td
class="even">{$tasks[t].ES}</td> <td class="even">{$tasks[t].ECT}</td> <td
class="even">{$tasks[t].LF}</td> <td class="even">{$tasks[t].activityslack}</td> <td
class="even"><a class="link" href="tiki-
admin_project_tasks.php?projectID={$projectID}&remove={$tasks[t].taskID}"></a> <a class="link" href="tiki-
admin_project_tasks.php?projectID={$projectID}&edit={$tasks[t].taskID}"></a></td> {/if} </tr> {/section} </table>

```

./templates/tiki-admin_projects.tpl

```

<H1>Projects</h1> <H2>Add project</h2> <form method="post" action="tiki-
admin_projects.php"> <table> <tr> <td> Projectname (unique): </td><td> <input type="text"
name="projectname" value="{ $projectdata.projectName}" {if
$projectdata.projectName}readonly{/if} size="30" maxlength="160" /> </td> </tr><tr> <td>
Description: </td><td> <textarea name="description" rows="5"
cols="80">{$projectdata.description}</textarea> </td> </tr><tr> <td> Starting date: </td><td>
<input type="text" name="start" value="{ $projectdata.start}" size="19" maxlength="19" /> </td>
</tr><tr> <td> Finishing date (estimated): </td><td> <input type="text" name="finish"
value="{ $projectdata.finish}" size="19" maxlength="19" /> </td> </tr> </table> <input
type="submit" name="{ $send}" value="add/edit project" /> </form> <H2>Tasks</h2> <table
class="normal" width="100%"> <tr> <td class="heading">ID</td> <td class="heading">name</td>
<td class="heading">start</td> <td class="heading">finish</td> <td class="heading">action</td>
</tr> {section name=t loop=$projects} <tr> {if $smarty.section.t.index % 2} <td
class="odd">{$projects[t].projectID}</td> <td class="odd">{$projects[t].projectName}</td> <td
class="odd">{$projects[t].start}</td> <td class="odd">{$projects[t].finish}</td> <td
class="odd"><a class="link" href="tiki-admin_projects.php?remove={$projects[t].projectID}"></a> <a class="link" href="tiki-
admin_projects.php?projectID={$projects[t].projectID}"></a> <a class="link" href="tiki-
admin_project_tasks.php?projectID={$projects[t].projectID}"></a></td> {else} <td class="even">{$projects[t].projectID}</td> <td
class="even">{$projects[t].projectName}</td> <td class="even">{$projects[t].start}</td> <td
class="even">{$projects[t].finish}</td> <td class="even"><a class="link" href="tiki-
admin_projects.php?remove={$projects[t].projectID}"></a> <a class="link" href="tiki-
admin_projects.php?projectID={$projects[t].projectID}"></a> <a class="link" href="tiki-
admin_project_tasks.php?projectID={$projects[t].projectID}"></a></td> {/if} </tr> {/section} </table>

```

```
<? require_once('tiki-setup.php'); require_once('lib/graphviz/GraphViz.php');
require_once('lib/projects/project_graphlib.php'); $garg = array(); if (!isset($_REQUEST['projectID'])) {
$smarty->assign('msg', "You need to define a project first"); $smarty->display("error.tpl"); die; } if
(!isset($_REQUEST['taskName'])) { $_REQUEST['taskName']=""; } $projgraph = new
ProjectGraphLib($dbTiki); $str=$projgraph->project_get_structure($_REQUEST['projectID'],
$_REQUEST['taskName']); $graph = new Image_GraphViz(); $projgraph->project_graph($str, $graph,
$garg); $graph->image() ?>
```

```
<? require_once('tiki-setup.php'); require_once('lib/projects/projectlib.php'); if ($tiki_p_admin != 'y') {
$smarty->assign('msg', tra("You dont have permission to use this feature"));
$smarty->display("error.tpl"); die; } $projectslib = new ProjectsLib($dbTiki); if
(isset($_REQUEST['addproject'])) { $projectslib->new_project($_REQUEST["projectname"],
$_REQUEST["description"], $_REQUEST["start"], $_REQUEST["finish"]); } elseif
(isset($_REQUEST['editproject'])) { $projectslib->edit_project($_REQUEST["projectname"],
$_REQUEST["description"], $_REQUEST["start"], $_REQUEST["finish"]); }
if(isset($_REQUEST['remove'])) { $projectslib->remove_project($_REQUEST["remove"]); }
if(isset($_REQUEST['projectID'])) { $projectdata =
$projectslib->get_projectdata($_REQUEST['projectID']); $smarty->assign('send', 'editproject'); } else {
$smarty->assign('send', 'addproject'); } $smarty->assign_by_ref('projectdata', $projectdata); $projects
= $projectslib->get_projects(); $smarty->assign_by_ref('projects',$projects); $smarty
->assign('mid','tiki-admin_projects.tpl'); $smarty->display('tiki.tpl'); ?>
```

```
<? require_once('tiki-setup.php'); require_once('lib/graphviz/GraphViz.php');
require_once('lib/projects/datetimelib.php'); require_once('lib/projects/project_tasklib.php');
require_once('lib/projects/project_graphlib.php'); if ($tiki_p_admin != 'y') { $smarty->assign('msg',
tra("You dont have permission to use this feature")); $smarty->display("error.tpl"); die; } if
(!isset($_REQUEST["projectID"])) { $smarty->assign('msg', "You need to define a project first");
$smarty->display("error.tpl"); die; } else { $projectID = $_REQUEST["projectID"]; }
$smarty->assign('projectID', $projectID); $projecttasklib = new ProjectTasksLib($dbTiki); if
(isset($_REQUEST['calculate_cpchart'])) { $projecttasklib->calculate_chart_forward($projectID, "", 0,
0); $projecttasklib->calculate_chart_backward($projectID, "", 0, 0); } if (isset($_REQUEST['addtask'])) {
$taskid = $projecttasklib->new_task($projectID, $_REQUEST["taskname"], $_REQUEST["description"],
$_REQUEST["ect"], $_REQUEST["ect_opt"], $_REQUEST["ect_pes"], $_REQUEST["crash_ect"],
$_REQUEST["es"], $_REQUEST["est_cost"], $_REQUEST["crash_cost"]); if
(isset($_REQUEST["predecessors"])) { foreach(array_keys($_REQUEST["predecessors"]) as $predid) {
$projecttasklib->add_predecessor($taskid, $_REQUEST["predecessors"][$$predid]); } } } elseif
(isset($_REQUEST['edittask'])) { $projecttasklib->edit_task($_REQUEST["taskid"],
$_REQUEST["taskname"], $_REQUEST["description"], $_REQUEST["ect"], $_REQUEST["ect_opt"],
$_REQUEST["ect_pes"], $_REQUEST["crash_ect"], $_REQUEST["es"], $_REQUEST["est_cost"],
$_REQUEST["crash_cost"]); $projecttasklib->remove_predecessors($_REQUEST["taskid"]); if
(isset($_REQUEST["predecessors"])) { foreach(array_keys($_REQUEST["predecessors"]) as $predid) {
$projecttasklib->add_predecessor($_REQUEST["taskid"], $_REQUEST["predecessors"][$$predid]); } }
} if(isset($_REQUEST['remove'])) { $projecttasklib->remove_task($_REQUEST["remove"]); }
```

```
if(isset($_REQUEST['edit'])) { $taskdata = $projecttasklib->get_taskdata($_REQUEST['edit']);  
$smarty->assign('send', 'edittask'); } else { $taskdata['predecessors']=array(); $smarty->assign('send',  
'addtask'); } $smarty->assign_by_ref('taskdata', $taskdata); $tasks =  
$projecttasklib->get_tasks($projectID); $smarty->assign_by_ref('tasks',$tasks); $smarty  
->assign('mid','tiki-admin_project_tasks.tpl'); $smarty ->display('tiki.tpl'); ?>
```

Don't mind this: KurtSysTestPage