

## LISTING PROGRAM

### 1.Login

#### a.Model User Login.php

```
<?php
/**
 * LoginForm class.
 * LoginForm is the data structure for keeping
 * user login form data. It is used by the 'login' action of
 * 'SiteController'.
 */
class UserLogin extends CFormModel
{
    public          $username;
    public $password;
    public $rememberMe;
    /**
     * Declares the validation rules.
     * The rules state that username and password are required,
     * and password needs to be authenticated.
     */
    public function rules()
    {
        return array(
            // username and password are required
            array('username, password', 'required'),          //
            rememberMe needs to be a boolean
            array('rememberMe', 'boolean'),
            // password needs to be authenticated
            array('password', 'authenticate'),
        );
    }
    /**
     * Declares attribute labels.
     */
    public function attributeLabels()
    {
        return array(
            'rememberMe'=>UserModule::t("Remember me next time"),
            'username'=>UserModule::t("username or email"),
            'password'=>UserModule::t("password"),
        );
    }
    /**
     * Authenticates the password.
     * This is the 'authenticate' validator as declared in rules().
     */
    public function authenticate($attribute,$params)
    {
        if(!$this->hasErrors()) // we only want to authenticate when no
input errors
        {
            $identity=new UserIdentity($this->username,$this->password);
            $identity->authenticate();
            switch($identity->errorCode)
```

```

{
case UserIdentity::ERROR_NONE:
$duration=$this->rememberMe ?
Yii::app()->controller->module->rememberMeTime : 0;
Yii::app()->user-
>login($identity,$duration);
break;
case UserIdentity::ERROR_EMAIL_INVALID:
$this->addError("username",UserModule::t("Email is incorrect.));
break;
case UserIdentity::ERROR_USERNAME_INVALID:
$this->addError("username",UserModule::t("Username          is
incorrect.));
break;
case UserIdentity::ERROR_STATUS_NOTACTIV:
$this->addError("status",UserModule::t("You      account      is      not
activated.));
break;
case UserIdentity::ERROR_STATUS_BAN:
$this->addError("status",UserModule::t("You          account          is
blocked.));
break;
case UserIdentity::ERROR_PASSWORD_INVALID:
$this->addError("password",UserModule::t("Password          is
incorrect.));
break;
}
}
}
}
}
}

```

## **b.View login.php**

```

<h1><?php echo UserModule::t("Login"); ?></h1>

<?phpif(Yii::app()->user>hasFlash('loginMessage')):
?><div class="success">
<?php echo Yii::app()->user->getFlash('loginMessage'); ?></div>

<?php endif;
?><div class="form">
<?php echo CHtml::beginForm();
?><p class="note"><?php echo UserModule::t('Isi NIM/NIP tanpa
spasi'); ?></p>

    <?php echo CHtml::errorSummary($model); ?>

    <div class="row">
        <?php echo CHtml::activeLabelEx($model,'NIM/NIP '); ?>
        <?php echo CHtml::activeTextField($model,'username')
?>
    </div>

    <div class="row">
        <?php echo CHtml::activeLabelEx($model,'Password ');
?>
        <?php echo

```

```

CHtml::activePasswordField($model, 'password')
?></div>
<div class="row submit">
    <?php echo
CHtml::submitButton(UserModule::t("Login"));
?></div>
<div class="row">
<p class="hint">
<?php //echo CHtml::link(UserModule::t("Lupa
Password?"),Yii::app()->getModule('user')>recoveryUrl);
?><?php echo
CHtml::link(UserModule::t("Daftar"),Yii::app()->getModule('user')-
>registrationUrl); ?></p>
</div>
<div class="row rememberMe">
<?php /*echo CHtml::activeCheckBox($model, 'rememberMe'); ?>
<?php echo CHtml::activeLabelEx($model, 'rememberMe');
*/?>
</div>
<?php echo CHtml::endForm(); ?>
</div><!-- form -->
<?php
$form = new CForm(array(
    'elements'=>array(
        'username'=>array(
            'type'=>'text',
            'maxlength'=>32,
        ),
        'password'=>array(
            'type'=>'password',
            'maxlength'=>32,
        ),
        'rememberMe'=>array(
            'type'=>'checkbox',
        )
    ),
    'buttons'=>array(
        'login'=>array(
            'type'=>'submit',
            'label'=>'Login',
        ),
    ),
    $model);
?>

```

### **c. Controller**

#### **LoginController.php**

```

<?php
class LoginController extends Controller
{
    public $defaultAction = 'login';

    /**
     * Displays the login page
     */
    public function actionLogin()
    {
        if (Yii::app()->user->isGuest) {

```

```

$model=new UserLogin;           //
collect user input data
if(isset($_POST['UserLogin']))
{
$model->attributes=$_POST['UserLogin'];
// validate user input and redirect to previous page if valid
if($model->validate()) {           $this-
>lastVisit();
if (Yii::app()->user->returnUrl=='/index.php')
$this->redirect(Yii::app()-
>controller->module->returnUrl);
else
$this->redirect(Yii::app()->user->returnUrl);
}
}
// display the login form
$this-
>render('/user/login',array('model'=>$model));
} else
$this->redirect(Yii::app()->controller->module-
>returnUrl);
}
private function lastVisit() {
    $lastVisit      =      User::model()->notsafe()-
>findByPk(Yii::app()->user->id);
    $lastVisit->lastvisit = time();
    $lastVisit->save();
}
}

```

## 2.Registrasi a.Model RegistrationForm.php

```

<?php
/**
 * RegistrationForm class.
 * RegistrationForm is the data structure for keeping
 * user registration form data. It is used by the 'registration'
 * action of 'UserController'.
 */
class RegistrationForm extends User {
public $verifyPassword;
public $verifyCode;
public function rules() {
    $rules = array(
        array('username, password, verifyPassword, email', 'required'),
        array('username', 'length', 'max'=>20, 'min' =>
3, 'message' => UserModule::t("Incorrect username (length between 3
and 20 characters).")),
        array('password', 'length', 'max'=>128, 'min' =>
4, 'message' => UserModule::t("Incorrect password (minimal length 4
symbols).")),
        array('email', 'email'),
        array('username', 'unique', 'message' =>
UserModule::t("This user's name already exists.")),
        array('email', 'unique', 'message' =>
UserModule::t("This user's email address already exists.")),
        //array('verifyPassword', 'compare',

```

```

'compareAttribute'=>'password',          'message'          =>
UserModule::t("Retype Password is incorrect.")),
array('username',      'match',      'pattern'      =>      '/^[A-Za-z0-
9_]+$\/u', 'message'      =>      UserModule::t("Incorrect symbols (Az0-
9).")),
);
if (!(isset($_POST['ajax']) &&
$_POST['ajax']==='registration-form')) {
array_push($rules,array('verifyCode', 'captcha',
'allowEmpty'=>!UserModule::doCaptcha('registration')));
}
array_push($rules,array('verifyPassword', 'compare',
'compareAttribute'=>'password', 'message' => UserModule::t("Retype
Password is incorrect."))); return $rules;
}
}

```

### b.View registration.php

```

<?php $this->pageTitle=Yii::app()->name . ' -
'.UserModule::t("Registration");
$this->breadcrumbs=array(
UserModule::t("Registration"),
);
?>
<h1><?php echo UserModule::t("Registration"); ?></h1>
<?php if (Yii::app()->user->hasFlash('registration')): ?>
<div class="success">
<?php echo Yii::app()->user->getFlash('registration'); ?>
</div>
<?php else: ?>
<div class="form">
<?php $form=$this->beginWidget('UActiveForm', array(
'id'=>'registration-form',
'enableAjaxValidation'=>true,
'disableAjaxValidationAttributes'=>array('RegistrationForm_v
erifyCode'),
'clientOptions'=>array(
'validateOnSubmit'=>true,
),
'htmlOptions' => array('enctype'=>'multipart/form-data'),
)); ?>
<p class="note"><?php echo UserModule::t('Fields with <span
class="required">*</span> are required. '); ?></p>
<?php echo $form->errorSummary(array($model,$profile)); ?>
<div class="row">
<?php echo $form->labelEx($model,'username'); ?>
<?php echo $form->textField($model,'username'); ?>
<?php echo $form->error($model,'username'); ?>
</div>
<div class="row">
<?php echo $form->labelEx($model,'password'); ?>
<?php echo $form->passwordField($model,'password'); ?>
<?php echo $form->error($model,'password'); ?>
<p class="hint">
<?php echo UserModule::t("Minimal password length 4 symbols.");
?>

```

```

</p>
</div>
<div class="row">
<?php echo $form->labelEx($model,'verifyPassword'); ?>
    <?php echo $form->passwordField($model,'verifyPassword'); ?>
<?php echo $form->error($model,'verifyPassword'); ?>
</div>
.
    <div class="row">
<?php echo $form->labelEx($model,'email');
?><?php echo $form->textField($model,'email');
?><?php echo $form->error($model,'email');
?></div>
<?php
$profileFields=$profile->getFields();          if
($profileFields) { foreach($profileFields as $field) {
?><div class="row">
<?php echo $form->labelEx($profile,$field->varname);
?><?php if ($widgetEdit = $field->widgetEdit($profile)) { echo
$widgetEdit; } elseif ($field->range) {
echo $form->dropDownList($profile,$field-
>varname,Profile::range($field->range));
} elseif ($field->field_type=="TEXT") {
echo$form->textArea($profile,$field-
>varname,array('rows'=>6, 'cols'=>50));
} else {
echo $form->textField($profile,$field-
>varname,array('size'=>60,'maxlength'=>(($field-
>field_size)?$field->field_size:255))); }
?><?php echo $form->error($profile,$field->varname);
?></div>
<?php
}
}
?><?php if (UserModule::doCaptcha('registration')):
?><div class="row">
<?php echo $form->labelEx($model,'verifyCode');
?><?php $this->widget('CCaptcha');
?><?php echo $form->textField($model,'verifyCode');
?><?php echo $form->error($model,'verifyCode');
?><p class="hint"><?php echo UserModule::t("Please enter the
letters as they are shown in the image above."); ?> <br/><?php
echo UserModule::t("Letters are not casesensitive."); ?></p>
</div>
<?php endif;
?><div class="row submit">
    <?php echo
CHtml::submitButton(UserModule::t("Register"));
?></div>
<?php $this->endWidget();
?></div><!-- form -->
<?php endif;
?>

```

### **c. Controller RegistrationController.php**

```

<?php
class RegistrationController extends Controller

```

```

{
public $defaultAction = 'registration';

/**
 * Declares class-based actions.
 */
public function actions()
{
return array(
'captcha'=>array(
'class'=>'CCaptchaAction',
'backColor'=>0xFFFFFFFF,
),
);
}
/**
 * Registration user
 */
public function actionRegistration() {
$model = new RegistrationForm;
$profile=new Profile;
$profile->regMode = true;
// ajax validator
if(isset($_POST['ajax']) &&
$_POST['ajax']==='registration-form')
{
echo
UActiveForm::validate(array($model,$profile));
Yii::app()->end();
}
if (Yii::app()->user->id) {
$this->redirect(Yii::app()->controller->module>profileUrl);
} else {
if(isset($_POST['RegistrationForm'])) {
$model->attributes=$_POST['RegistrationForm'];
$profile->attributes=(isset($_POST['Profile'])?$_POST['Profile']:a
rray()); ;
if($model->validate() &&$profile-
>validate())
{
$soucePassword = $model-
$password;
$model-
>activkey=UserModule::encrypting(microtime().$model->password);
$model->password=UserModule::encrypting($model->password);
$model->verifyPassword=UserModule::encrypting($model-
>verifyPassword);
$model->superuser=0;
$model->status=((Yii::app()->controller->module-
>activeAfterRegister)?User::STATUS_ACTIVE:User::STATUS_NOACTIVE);
if ($model->save()) {
$profile-
$profile->save();
if (Yii::app()-

```

```

>controller->module->sendActivationMail) {
$activation_url =
$this-
>createAbsoluteUrl('/user/activation/activation',array("activkey"
=> $model->activkey, "email" => $model->email));

UserModule::sendMail($model->email,UserModule::t("You   registered
from
{site_name}",array('{site_name}'=>Yii::app()->name)),UserModule::t(
"Please activate you account go to
{activation_url}",array('{activation_url}'=>$activation_url)));
}
if ((Yii::app()->controller->module->loginNotActiv||(Yii::app()-
>controller->module->activeAfterRegister&&Yii::app()->controller-
>module-
>sendActivationMail==false))&&Yii::app()->controller->module-
>autoLogin) {
    $identity=new UserIdentity($model->username,$sourcePassword);
    $identity-
    >authenticate();
    Yii::app()-
    >user->login($identity,0);
    $this-
    >redirect(Yii::app()->controller->module->returnUrl);
} else {
    if
    (!Yii::app()->controller->module-
    >activeAfterRegister&&!Yii::app()->controller->module-
    >sendActivationMail) {
        Yii::app()->user->setFlash('registration',UserModule::t("Thank you
for your registration. Contact Administrator to activate your
account."));
    }
    elseif(Yii::app()->controller->module-
    >activeAfterRegister&&Yii::app()->controller->module-
    >sendActivationMail==false) {
        Yii::app()->user->setFlash('registration',UserModule::t("Thank you
for
your
registration.
Please
{{login}}.",array('{{login}}'=>CHtml::link(UserModule::t('Login'),
Yii::app()->controller->module->loginUrl))));
    }
    elseif(Yii::app()->controller->module->loginNotActiv) {
        Yii::app()->user->setFlash('registration',UserModule::t("Thank you
for your registration. Please check your email or login."));
    } else {
        Yii::app()->user->setFlash('registration',UserModule::t("Thank you
for your registration. Please check your email."));
    }
    $this->refresh();
}
}
} else $profile->validate();
}
$this-
>render('/user/registration',array('model'=>$model,'profile'=>$pro
file));
}
}
}
}

```

### 3. Judul a. Model Mahasiswa.php

```
<?php

/**
 * This is the model class for table "mahasiswa".
 * *
 * The followings are the available columns in table 'mahasiswa':
 * @property integer $id_mhs
 * @property string $nim_mhs
 * @property string $nama_mhs
 * @property string $fokus_studi
 * @property string $judul_skripsi
 * @property string $pembimbing
 * @property string $tgl_masuk
 * @property string $tgl_ujian
 * @property string $pengujil
 * @property string $penguji2
 */
class Mahasiswa extends CActiveRecord
{
/**
 * @return string the associated database table name
 */
public function tableName()
{
return 'mahasiswa';
}
/**
 * @return array validation rules for model attributes.
 */
public function rules()
{
// NOTE: you should only define rules for those attributes that
// will receive user inputs.
return array(
array('nim_mhs, nama_mhs, fokus_studi, judul_la, pembimbing,
tgl_masuk', 'required'),
array('nim_mhs', 'length', 'max'=>11), array('nama_mhs,
pembimbing, pengujil, penguji2', 'length', 'max'=>50),
array('fokus_studi', 'length', 'max'=>10),
array('judul_la', 'length', 'max'=>250),
array('tgl_ujian', 'safe'),
// The following rule is used by search(). //
@todo Please remove those attributes that should not be
searched.
array('id_mhs, nim_mhs, nama_mhs, fokus_studi, judul_la,
pembimbing, tgl_masuk, tgl_ujian, pengujil, penguji2', 'safe',
'on'=>'search'),
);
}
/**
 * @return array relational rules.
 */
public function relations()
{
// NOTE: you may need to adjust the relation name and the related
```

```

// class name for the relations automatically generated below.
return array(
);
}
/**
 * @return array customized attribute labels (name=>label)
 */ public function attributeLabels()
{
return array( 'id_mhs' => 'Id Mahasiswa',
              'nim_mhs' => 'NIM',
              'nama_mhs' => 'Nama Mahasiswa',
              'fokus_studi' => 'Fokus Studi',
              'judul_la' => 'Judul Laporan Akhir',
              'pembimbing' => 'Pembimbing',
              'tgl_masuk' => 'Tanggal Masuk',
              'tgl_ujian' => 'Tanggal Ujian',
              'penguji1' => 'Penguji 1',
              'penguji2' => 'Penguji 2',
);
}
/**
 * Retrieves a list of models based on the current search/filter
 * conditions.
 *
 * Typical usecase:
 * - Initialize the model fields with values from filter form.
 * - Execute this method to get CActiveDataProvider instance which
 *   will filter
 * models according to data in model fields. * - Pass data
 * provider to CGridView, CListView or any similar widget.
 *
 * @return CActiveDataProvider the data provider that can return
 * the models
 * based on the search/filter conditions.
 */ public function search()
{
// @todo Please modify the following code to remove attributes
// that should not be searched.

$criteria=new CDbCriteria;
$criteria->compare('id_mhs',$this->id_mhs);
$criteria->compare('nim_mhs',$this->nim_mhs,true);
$criteria->compare('nama_mhs',$this->nama_mhs,true);
$criteria->compare('fokus_studi',$this->fokus_studi,true);
$criteria->compare('judul_la',$this->judul_la,true);
$criteria->compare('pembimbing',$this->pembimbing,true);
$criteria->compare('tgl_masuk',$this->tgl_masuk,true);
$criteria->compare('tgl_ujian',$this->tgl_ujian,true);
$criteria->compare('penguji1',$this->penguji1,true);
$criteria->compare('penguji2',$this->penguji2,true);
return new CActiveDataProvider($this, array(
'criteria'=>$criteria,
));
}

```

```

/**
 * Returns the static model of the specified AR class. * Please
 * note that you should have this exact method in all your
 * CActiveRecord descendants!
 * @param string $className active record class name.
 * @return Mahasiswa the static model class
 */ public static function model($className=__CLASS__)
{
return parent::model($className);
}
}

```

## **b.View admin.php**

### **(Daftar Judul)**

```

<?php
/* @var $this MahasiswaController */
/* @var $model Mahasiswa */
$this->breadcrumbs=array(
'Mahasiswas'=>array('index'),
'Manage',
);
$this->menu=array(
//array('label'=>'List Mahasiswa', 'url'=>array('index')),
array('label'=>'Tambah Daftar',
'url'=>array('create'),'visible'=>Yii::app()->user-
>checkAccess('Staff')),
array('label'=>'Tambah Daftar',
'url'=>array('create'),'visible'=>Yii::app()->user-
>checkAccess('Kaprodi')),
);
Yii::app()->clientScript->registerScript('search', "
$('.search button').click(function(){
$('.search-form').toggle(); return
false;
});
$('.search-form form').submit(function(){ $('#mahasiswa-
grid').yiiGridView('update', {
data: $(this).serialize()
});
return false;
});
");
?>
<h1>Daftar Judul Laporan Akhir</h1>
<?php $this->widget('application.extensions.tablesorter.Sorter',
array(
'data'=>$records,
'columns'=>array(
'nim_mhs',
'nama_mhs',
'judul_skripsi',
'tgl_masuk',
'tgl_ujian',
/*
'id_mhs',

```

```

'fokus_studi',
'pembimbing',
'penguji1',
'penguji2',
*/
),
)); ?>
create.php
<?php
/* @var $this MahasiswaController */
/* @var $model Mahasiswa */
$this->breadcrumbs=array(
'Mahasiswas'=>array('index'),
'Create',
);
$this->menu=array(
//array('label'=>'List Mahasiswa', 'url'=>array('index')),
array('label'=>'Daftar Laporan Akhir', 'url'=>array('admin')),
);
?>
<h1>Tambah Judul Laporan Akhir</h1>
<?php $this->renderPartial('_form', array('model'=>$model)); ?>
_form.php
<?php
/* @var $this MahasiswaController */
/* @var $model Mahasiswa */
/* @var $form CActiveForm */
?>
<div class="form">
<?php $form=$this->beginWidget('CActiveForm', array(
'id'=>'mahasiswa-form',
// Please note: When you enable ajax validation, make sure the
corresponding
// controller action is handling ajax validation correctly.
// There is a call to performAjaxValidation() commented in
generated controller code.
// See class documentation of CActiveForm for details on this.
'enableAjaxValidation'=>false,
)); ?>
<p class="note">Fields dengan tanda <span
class="required">*</span> harus diisi.</p>
<?php echo $form->errorSummary($model); ?>
<div class="row">
<?php echo $form->labelEx($model, 'nim_mhs'); ?>
<?php echo $form-
>textField($model, 'nim_mhs', array('size'=>11, 'maxlength'=>11)); ?>
<?php echo $form->error($model, 'nim_mhs'); ?>
</div>
<div class="row">
<?php echo $form->labelEx($model, 'nama_mhs'); ?>
<?php echo $form->textField($model, 'nama_mhs', array('size'=>50, 'maxlength'=>50));
?>
<?php echo $form->error($model, 'nama_mhs'); ?>
</div>
<div class="row">

```

```

<?php echo $form->labelEx($model,'fokus_studi'); ?>
<?php
    echo
    $form-
>dropDownList($model,'fokus_studi',array('aljabar'=>'Aljabar','geo
metri'=>'Geometri','komputer'=>'Komputer',
'statistika'=>'Statistika','terapan'=>'Terapan','pendidikan'=>'Pen
didikan')); ?>
<?php echo $form->error($model,'fokus_studi'); ?>
</div>
<div class="row">
<?php echo $form->labelEx($model,'judul_skripsi'); ?> <?php echo
$form-
>textField($model,'judul_skripsi',array('size'=>60,'maxlength'=>25
0)); ?>
<?php echo $form->error($model,'judul_skripsi'); ?>
</div>
<div class="row">
<?php echo $form->labelEx($model,'pembimbing'); ?>
<?php
    echo
    $form-
>textField($model,'pembimbing',array('size'=>50,'maxlength'=>50));
?>
<?php echo $form->error($model,'pembimbing'); ?>
</div>
<div class="row">
<?php echo $form->labelEx($model,'tgl_masuk'); ?> <?php
$this->widget('zii.widgets.jui.CJuiDatePicker',
array(
// you must specify name or model/attribute
'model'=>$model,
'attribute'=>'tgl_masuk',
'value' => $model->tgl_masuk,
'options' => array(
'dateFormat'=>'yy-mm-dd',
'showOn'=>'button',
'yearRange'=>'-60',
'changeMonth' => 'true',
'changeYear' => 'true',
'constrainInput' => 'false',
'duration'=>'fast',
'showAnim' =>'slide',
// 'showAnim' =>'fold',
),
'htmlOptions'=>array('size'=>20,),
); ?>
<?php echo $form->error($model,'tgl_masuk'); ?>
</div>
<div class="row">
<?php echo $form->labelEx($model,'tgl_ujian'); ?>
<?php $this->widget('zii.widgets.jui.CJuiDatePicker',
array(
// you must specify name or model/attribute
'model'=>$model,
'attribute'=>'tgl_ujian',
'value' => $model->tgl_ujian,
'options' => array(
'dateFormat'=>'yy-mm-dd',

```

```

'showOn'=>'button',
'yearRange'=>'-60',
'changeMonth' => 'true',
'changeYear' => 'true',
'constrainInput' => 'false',
'duration'=>'fast',
'showAnim' =>'slide',
// 'showAnim' =>'fold',
),
'htmlOptions'=>array('size'=>20,),
)
); ?>
<?php echo $form->error($model,'tgl_ujian'); ?>
</div>
<div class="row">
<?php echo $form->labelEx($model,'penguji1'); ?>
<?php echo $form-
>textField($model,'penguji1',array('size'=>50,'maxlength'=>50));
?>
<?php echo $form->error($model,'penguji1'); ?>
</div>
<div class="row">
<?php echo $form->labelEx($model,'penguji2'); ?>
<?php echo $form-
>textField($model,'penguji2',array('size'=>50,'maxlength'=>50));
?>
<?php echo $form->error($model,'penguji2'); ?>
</div>
<div class="row buttons">
<?php echo CHtml::submitButton($model->isNewRecord ?
'Tambah' : 'Save'); ?>
</div>
<?php $this->endWidget(); ?>
</div><!-- form -->

```

### view.php

```

<?php
/* @var $this MahasiswaController */
/* @var $model Mahasiswa */
$this->breadcrumbs=array(
'Mahasiswas'=>array('index'),
$model->id_mhs,
);
$this->menu=array(
//array('label'=>'List Mahasiswa', 'url'=>array('index')),
array('label'=>'Tambah Daftar',
'url'=>array('create'),'visible'=>Yii::app()->user-
>checkAccess('Staff')),
array('label'=>'Tambah Daftar',
'url'=>array('create'),'visible'=>Yii::app()->user-
>checkAccess('Kaprodi')), array('label'=>'Update
Data', 'url'=>array('update',
'id'=>$model->id_mhs),'visible'=>Yii::app()->user-
>checkAccess('Staff')), array('label'=>'Update
Data', 'url'=>array('update',
'id'=>$model->id_mhs),'visible'=>Yii::app()->user-

```

```

>checkAccess('Kaprodi')),
array('label'=>'Hapus Data', 'url'=>'#',
'linkOptions'=>array('submit'=>array('delete','id'=>$model-
>id_mhs), 'confirm'=>'Anda yakin?'), 'visible'=>Yii::app()->user-
>checkAccess('Staff')),
array('label'=>'Hapus Data', 'url'=>'#',
'linkOptions'=>array('submit'=>array('delete','id'=>$model-
>id_mhs), 'confirm'=>'Anda yakin?'), 'visible'=>Yii::app()->user-
>checkAccess('Kaprodi')),
array('label'=>'Daftar Judul Skripsi',
'url'=>array('admin')),
);
?>
<h1>Detail Judul #<?php echo $model->nim_mhs; ?></h1>
<?php $this->widget('zii.widgets.CDetailView', array(
'data'=>$model,
'attributes'=>array(
'nim_mhs',
'nama_mhs',
'fokus_studi',
'judul_la',
'pembimbing',
'tgl_masuk',
'tgl_ujian',
'pengujil',
'penguji2',
),
)); ?>
_view.php
<?php
/* @var $this MahasiswaController */
/* @var $data Mahasiswa */
?>
<div class="view">
<b><?php echo CHtml::encode($data-
>getAttributeLabel('nim_mhs')); ?></b>
<?php echo CHtml::link(CHtml::encode($data->nim_mhs),
array('view', 'id'=>$data->id_mhs)); ?>
<br />
<b><?php echo CHtml::encode($data->getAttributeLabel('nama_mhs'));
?></b>
<?php echo CHtml::encode($data->nama_mhs); ?>
<br />
<b><?php
echo
CHtml::encode($data->getAttributeLabel('fokus_studi')); ?></b>
<?php echo CHtml::encode($data->fokus_studi); ?>
<br />
<b><?php echo CHtml::encode($data-
>getAttributeLabel('judul_skripsi')); ?></b>
<?php echo CHtml::encode($data->judul_skripsi); ?>
<br />
<b><?php
echo
CHtml::encode($data->getAttributeLabel('pembimbing')); ?></b>
<?php echo CHtml::encode($data->pembimbing); ?>
<br />

```

```

<b><?php
CHtml::encode($data>getAttributeLabel('tgl_masuk')); ?></b>      echo
<?php echo CHtml::encode($data->tgl_masuk); ?>
<br />
<?php /*
<b><?php
CHtml::encode($data>getAttributeLabel('tgl_ujian')); ?></b>      echo
<?php echo CHtml::encode($data->tgl_ujian); ?>
<br />
<b><?php echo CHtml::encode($data>getAttributeLabel('penguji1'));
?></b>
<?php echo CHtml::encode($data->penguji1); ?>
<br />
<b><?php echo CHtml::encode($data>getAttributeLabel('penguji2'));
?></b>
<?php echo CHtml::encode($data->penguji2); ?>
<br />
*/ ?>
</div>

```

### **c. Controller**

#### **MahasiswaController.php**

```

<?php
class MahasiswaController extends Controller
{
/**
 * @var string the default layout for the views. Defaults to
 * '//layouts/column2', meaning
 * using two-column layout. See
 * 'protected/views/layouts/column2.php'.
 */public $layout='//layouts/column2';
/**
 * @return array action filters
 */public function filters()
{
return array(
// 'rights',
'accessControl', // perform access control for CRUD operations
'postOnly + delete', // we only allow deletion via POST request
);
}
/**
 * Specifies the access control rules.
 * This method is used by the 'accessControl' filter.
 * @return array access control rules
 */public function accessRules()
{
return array(
array('allow', // allow authenticated user to perform 'create' and
'update' actions
'actions'=>array('admin','view','index'),'users'=>array('@',)),
array('allow', // allow admin user to perform 'admin' and 'delete'
actions 'actions'=>array('create','update','delete'),'hak
akses'=>array('SuperAdmin','Staff','Kaprodin')),
array('deny',
// deny all users 'users'=>array('*'),),);
}
/**

```

```

* Displays a particular model.
* @param integer $id the ID of the model to be displayed
*/public function actionView($id)
{
$this->render('view',array('model'=>$this->loadModel($id),));
}
/**
*Creates a new model.
*If creation is successful, the browser will be redirected to the
'view' page.
*/public function actionCreate()
{
$model=new Mahasiswa;
// Uncomment the following line if AJAX validation is needed
// $this->performAjaxValidation($model);
if(isset($_POST['Mahasiswa']))
{
$model->attributes=$_POST['Mahasiswa'];
if($model->save())
$this->redirect(array('view','id'=>$model->id_mhs));
}
$this->render('create',array( 'model'=>$model,
));
}
/**
* Updates a particular model.
* If update is successful, the browser will be redirected to the
'view' page.
* @param integer $id the ID of the model to be updated
*/public function actionUpdate($id)
{
$model=$this->loadModel($id);
// Uncomment the following line if AJAX validation is needed
// $this->performAjaxValidation($model);
if(isset($_POST['Mahasiswa']))
{
$model->attributes=$_POST['Mahasiswa'];
if($model->save())
$this->redirect(array('view','id'=>$model->id_mhs));
}
$this->render('update',array(
'model'=>$model,
));
}
/**
* Deletes a particular model.
* If deletion is successful, the browser will be redirected to
the 'admin' page.
* @param integer $id the ID of the model to be deleted
*/public function actionDelete($id)
{
$this->loadModel($id)->delete();
// if AJAX request (triggered by deletion via admin
grid view), we should not redirect the browser
if(!isset($_GET['ajax']))

```

```

$this->redirect(isset($_POST['returnUrl']) ?
$_POST['returnUrl'] : array('admin'));
}
/**
 * Lists all models.
 */public function actionIndex()
{
$dataProvider=new CActiveDataProvider('Mahasiswa');
$this->render('index',array(
'dataProvider'=>$dataProvider,
));
}
/**
 * Manages all models.
 */public function actionAdmin ()
{
$records=Mahasiswa::model()->findAll();
$this->render('admin',array(
'records'=>$records,
));
}
/**
 * Returns the data model based on the primary key given in the
GET variable.
 * If the data model is not found, an HTTP exception will be
raised.
 * @param integer $id the ID of the model to be loaded
 * @return Mahasiswa the loaded model
 * @throws CHttpException
 */public function loadModel($id)
{
$model=Mahasiswa::model()->findByPrimaryKey($id);
if($model===null)
throw new CHttpException(404,'The requested page
does not exist.');
```

#### **4. Proposal**

##### **a. Model**

```

<?php
/**
 * This is the model class for table "proposal".
```

```

* *
* The followings are the available columns in table 'proposal':
* @property integer $id_proposal
* @property string $nim_mhs
* @property string $judul_proposal
* @property string $file_proposal
*/class Proposal extends CActiveRecord
{
/**
* @return string the associated database table name
*/public function tableName()
{
return 'proposal';
}
/**
* @return array validation rules for model attributes.
*/public function rules()
{
// NOTE: you should only define rules for those attributes that
// will receive user inputs.
return array(
array('nim_mhs, judul_proposal', 'required'),
array('nim_mhs', 'length', 'max'=>20),
array('judul_proposal', 'length', 'max'=>250),
array('file_proposal', 'file', 'types'=>'doc,
docx, pdf'),
// The following rule is used by search(). //
@todo Please remove those attributes that should not be
searched.
array('id_proposal, nim_mhs, judul_proposal, file_proposal',
'safe', 'on'=>'search'),);
}
/**
* @return array relational rules.
*/public function relations()
{
// NOTE: you may need to adjust the relation name and the related
// class name for the relations automatically generated below.
return array(
);
}
/**
* @return array customized attribute labels (name=>label)
*/public function attributeLabels()
{
return array(
'id_proposal' => 'Id Proposal',
'nim_mhs' => 'Nim Mahasiswa',
'judul_proposal' => 'Judul Proposal',
'file_proposal' => 'File Proposal',
);
}
/**
* Retrieves a list of models based on the current search/filter
conditions.
*

```

```

* Typical usecase:
* - Initialize the model fields with values from filter form.
* - Execute this method to get CActiveDataProvider instance which
  will filter
* models according to data in model fields.      * - Pass data
  provider to CGridView, CListView or any similar widget.
*
* @return CActiveDataProvider the data provider that can return
  the models
* based on the search/filter conditions.
*/public function search()
{
// @todo Please modify the following code to remove attributes
that should not be searched.
$criteria=new CDbCriteria;
$criteria->compare('id_proposal',$this->id_proposal);
$criteria->compare('nim_mhs',$this->nim_mhs,true);
$criteria->compare('judul_proposal',$this->judul_proposal,true);
$criteria->compare('file_proposal',$this->file_proposal,true);
return new CActiveDataProvider($this, array(
'criteria'=>$criteria,
));
}
/**
* Returns the static model of the specified AR class.
* Please note that you should have this exact method in all
  your CActiveRecord descendants!
* @param string $className active record class name.
* @return Proposal the static model class
*/public static function model($className=__CLASS__)
{
return parent::model($className);
}
}

```

#### **b.View admin.php (List Proposal)**

```

<?php
/* @var $this ProposalController */
/* @var $model Proposal */

$this->breadcrumbs=array(
'Proposals'=>array('index'),
'Manage',
);
$this->menu=array(
//array('label'=>'List Proposal', 'url'=>array('index')),
//array('label'=>'Create Proposal', 'url'=>array('create')), );

Yii::app()->clientScript->registerScript('search', "
$('.search button').click(function(){
$('.search-form').toggle(); return
false;
});
$('.search-form form').submit(function(){ $('#proposal-
grid').yiiGridView('update', {

```

```

data: $(this).serialize()
});
return false;
});
");
?>
<h1>Daftar Proposal</h1>
<?php echo CHtml::link('Advanced
Search','#',array('class'=>'search-button')); ?>
<div class="search-form" style="display:none">
<?php $this->renderPartial('_search',array(
'model'=>$model,
)); ?>
</div><!-- search-form -->
<?php $this->widget('zii.widgets.grid.CGridView', array(
'id'=>'proposal-grid',
'dataProvider'=>$model->search(),
'filter'=>$model,
'columns'=>array(
'nim_mhs',
'judul_proposal',
array(
'name'=>'file_proposal',
'type'=>'raw',
'value'=>'CHtml::link($data->file_proposal,
"http://" . $_SERVER["SERVER_NAME"] . Yii::app()->request->baseUri
."/images/foto/" . $data->file_proposal)',
),
array(
'class'=>'CButtonColumn',
),
),
));
?>

```

#### **create.php (Upload Proposal)**

```

<?php
/* @var $this ProposalController */
/* @var $model Proposal */
$this->breadcrumbs=array(
'Proposals'=>array('index'),
'Create',
);
$this->menu=array(
//array('label'=>'List Proposal', 'url'=>array('index')),
//array('label'=>'Manage Proposal', 'url'=>array('admin')), );
?>
<h1>Upload Proposal</h1>
<?php $this->renderPartial('_form', array('model'=>$model)); ?>

```

#### **\_form.php**

```

<?php
/* @var $this ProposalController */
/* @var $model Proposal */ /*
@var $form CActiveForm */
?>

```

```

<div class="form">
<?php $form=$this->beginWidget('CActiveForm', array(
'id'=>'proposal-form',
// Please note: When you enable ajax validation, make sure the
corresponding
// controller action is handling ajax validation correctly.
// There is a call to performAjaxValidation() commented in
generated controller code.
// See class documentation of CActiveForm for details on this.
'enableAjaxValidation'=>false,
'htmlOptions'=>array('enctype'=>'multipart/form-data'),
)); ?>
<p class="note">Fields dengan tanda <span
class="required">*/</span> harus diisi.</p>
<?php echo $form->errorSummary($model); ?>
<div class="row">
<?php echo $form-
>hiddenField($model, 'nim_mhs', array('class'=>'User', 'value'=>Yii::
app()->user->name)); ?>
<?php echo $form->error($model, 'nim_mhs'); ?>
</div>
<div class="row">
<?php echo $form->labelEx($model, 'judul_proposal'); ?>
<?php echo $form-
>textArea($model, 'judul_proposal', array('size'=>60, 'maxlength'=>2
50, 'row'=>4, 'cols'=>250)); ?>
<?php echo $form->error($model, 'judul_proposal'); ?>
</div>
<div class="row">
<?php echo $form->labelEx($model, 'file_proposal'); ?> <?php echo
$form-
>fileField($model, 'file_proposal', array('size'=>30, 'maxlength'=>30
)); ?>
<?php echo $form->error($model, 'file_proposal'); ?>
</div>
<div class="row buttons">
<?php echo CHtml::submitButton($model->isNewRecord ?
'Upload' : 'Save'); ?>
</div>
<?php $this->endWidget(); ?>
</div><!-- form -->

```

### view.php

```

<?php
/* @var $this ProposalController */
/* @var $model Proposal */
$this->breadcrumbs=array(
'Proposals'=>array('index'),
$model->id_proposal,
);
$this->menu=array(
//array('label'=>'List Proposal',
'url'=>array('index'),'visible'=>Yii::app()->user-
>checkAccess('Kaprodi')),

```

```

//array('label'=>'Upload Proposal', 'url'=>array('create')),
array('label'=>'Hapus Proposal', 'url'=>'#',
'linkOptions'=>array('submit'=>array('delete','id'=>$model-
>id_proposal),'confirm'=>'Anda yakin?'),'visible'=>Yii::app()-
>user->checkAccess('Kaprodi')),
array('label'=>'Daftar Proposal',
'url'=>array('admin'),'visible'=>Yii::app()->user-
>checkAccess('Kaprodi')),
);
?>

```

```

<h1>Detail Proposal <?php echo $model->nim_mhs; ?></h1>
<?php $this->widget('zii.widgets.CDetailView', array(
'data'=>$model,
'attributes'=>array(
'nim_mhs',
'judul_proposal',
'file_proposal',
),
)); ?>

```

### **\_view.php**

```

<?php
/* @var $this ProposalController */
/* @var $data Proposal */
?>
<div class="view">
<b><?php echo CHtml::encode($data-
>getAttributeLabel('nim_mhs')); ?></b>
<?php echo CHtml::link(CHtml::encode($data->nim_mhs),
array('view', 'id'=>$data->id_proposal)); ?>
<br />
<b><?php echo CHtml::encode($data-
>getAttributeLabel('judul_proposal')); ?></b>
<?php echo CHtml::encode($data->judul_proposal); ?>
<br />
<b><?php echo CHtml::encode($data-
>getAttributeLabel('file_proposal')); ?></b>
<?php echo CHtml::link($data->file_proposal, "http://" .
$_SERVER["SERVER_NAME"] . Yii::app()->request->baseUrl .
"/images/foto/" . $data->file_proposal); ?>
<br />
</div>

```

### **c. Controller**

```

<?php
class ProposalController extends Controller
{
/**
 * @var string the default layout for the views. Defaults to
 * '//layouts/column2', meaning
 * using two-column layout. See
 * 'protected/views/layouts/column2.php'.
 * @public $layout='//layouts/column2';
 */
/**
 * @return array action filters
 * @public function filters()
 * {

```

```

return array(
    //'rights',
    'accessControl', // perform access control for
    CRUD operations
    'postOnly + delete', // we only allow deletion via POST request
);
}
/**
 * Specifies the access control rules.
 * This method is used by the 'accessControl' filter.
 * @return array access control rules
 */public function accessRules()
{
    return array(
        array('allow', // allow all users to perform
            'index' and 'view' actions
            'actions'=>array('admin','index','delete','view'),
            'hak akses'=>array('Kaprodi')),
        array('allow', // allow authenticated user to perform 'create' and
            'update' actions
            'actions'=>array('create','view'),
            'hak akses'=>array('Mahasiswa')),
        array('allow', // allow admin user to perform
            'admin' and 'delete' actions
            'actions'=>array('create','update','delete','admin','index',
            'view'),'hak akses'=>array('SuperAdmin'), ),
        array('deny', // deny all users'users'=>array('*'),), );
    }
}
/**
 * Displays a particular model.
 * @param integer $id the ID of the model to be displayed
 */public function actionView($id)
{
    $this->render('view',array(
        'model'=>$this->loadModel($id),
    ));
}
/**
 * Creates a new model.
 * If creation is successful, the browser will be redirected to
    the 'view' page.
 */public function actionCreate()
{
    $model=new Proposal;
    // Uncomment the following line if AJAX validation is needed
    // $this->performAjaxValidation($model);
    if(isset($_POST['Proposal']))
    {
        $model->attributes=$_POST['Proposal'];
        $model->file_proposal =
        CUploadedFile::getInstance($model, 'file_proposal');
        if($model->save())
        if(strlen($model->file_proposal)>0)
        $model->file_proposal->saveAs(Yii::app()-
        >basePath .'../images/foto/' .$model->file_proposal);
        $this->redirect(array('view','id'=>$model->

```

```

>id_proposal));
}
$this->render('create',array(
'model'=>$model,
));
}
/**
 * Updates a particular model.
 * If update is successful, the browser will be redirected to the
   'view' page.
 * @param integer $id the ID of the model to be updated
 */public function actionUpdate($id)
{
$model=$this->loadModel($id);
// Uncomment the following line if AJAX validation is needed
// $this->performAjaxValidation($model);
if(isset($_POST['Proposal']))
{
$model->attributes=$_POST['Proposal'];
$model->file_proposal =
CUploadedFile::getInstance($model, 'file_proposal');
if($model->save())
if(strlen($model->file_proposal)>0) $model->
>file_proposal->saveAs(Yii::app()-
>basePath '../images/foto/' .$model->file_proposal);
$this->redirect(array('view','id'=>$model->
id_proposal));
}
$this->render('update',array(
'model'=>$model,
));
}
/**
 * Deletes a particular model.
 * If deletion is successful, the browser will be redirected to
   the 'admin' page.
 * @param integer $id the ID of the model to be deleted
 */public function actionDelete($id)
{
$this->loadModel($id)->delete();
// if AJAX request (triggered by deletion via admin
grid view), we should not redirect the browser
if(!isset($_GET['ajax']))
$this->redirect(isset($_POST['returnUrl']) ?
$_POST['returnUrl'] : array('admin'));
}
/**
 * Lists all models.
 */public function actionIndex()
{
$dataProvider=new CActiveDataProvider('Proposal');
$this->render('index',array(
'dataProvider'=>$dataProvider,
));
}
/**

```

```

* Manages all models.
*/public function actionAdmin()
{
    $model=new Proposal('search');
    $model->unsetAttributes(); // clear any default
values
    if(isset($_GET['Proposal']))
        $model->attributes=$_GET['Proposal'];

    $this->render('admin',array(
        'model'=>$model,
    ));
}

/**
 * Returns the data model based on the primary key given in the
 * GET variable.
 * If the data model is not found, an HTTP exception will be
 * raised.
 * @param integer $id the ID of the model to be loaded
 * @return Proposal the loaded model
 * @throws CHttpException
 */public function loadModel($id)
{
    $model=Proposal::model()->findByPk($id);
    if($model===null)
        throw new CHttpException(404,'The requested page does not
        exist.');
```

return \$model;

```

}
/**
 * Performs the AJAX validation.
 * @param Proposal $model the model to be validated
 */protected function performAjaxValidation($model)
{
    if(isset($_POST['ajax']) &&
    $_POST['ajax']=== 'proposal-form')
    {
        echo CActiveForm::validate($model);
        Yii::app()->end();
    }
}
}
}
}

```

## 5. Dosen

### a. Model Dosen.php

```

<?php
/**
 * This is the model class for table "dosen".
 *
 * The followings are the available columns in table 'dosen':
 * @property integer $id_dosen
 * @property string $nama_dosen
 * @property string $nip
 * @property string $tgl_lahir
 * @property string $jabatan

```

```

* @property string $gelar
* @property string $pendidikan
* @property string $bidang_keahlian
* @property string $pangkat
*/
class Dosen extends CActiveRecord
{
/**
* @return string the associated database table name
*/public function tableName()
{
return 'dosen';
}
/**
* @return array validation rules for model attributes.
*/public function rules()
{
// NOTE: you should only define rules for those attributes that
// will receive user inputs.
return array(
array('nama_dosen, nip, tgl_lahir, jabatan, gelar, pendidikan,
bidang_keahlian, pangkat', 'required'),
array('nama_dosen', 'length', 'max'=>50),
array('nip, pangkat', 'length', 'max'=>30),
array('jabatan', 'length', 'max'=>20),
array('gelar', 'length', 'max'=>20), // The
following rule is used by search(). //
@todo Please remove those attributes that should not be
searched.
array('id_dosen, nama_dosen, nip, tgl_lahir, jabatan, gelar,
pendidikan, bidang_keahlian, pangkat', 'safe',
'on'=>'search'),
);
}
/**
* @return array relational rules.
*/public function relations()
{
// NOTE: you may need to adjust the relation name and the related
// class name for the relations automatically generated below.
return array(
);
}
/**
* @return array customized attribute labels (name=>label)
*/public function attributeLabels()
{
return array(
'id_dosen' => 'Id Dosen',
'nama_dosen' => 'Nama Dosen',
'nip' => 'NIP',
'tgl_lahir' => 'Tanggal Lahir',
'jabatan' => 'Jabatan',
'gelar' => 'Gelar',
'pendidikan' => 'Pendidikan',
'bidang_keahlian' => 'Bidang Keahlian',

```

```

'pangkat' => 'Pangkat',
);
}
/**
 * Retrieves a list of models based on the current search/filter
 * conditions.
 *
 * Typical usecase:
 * - Initialize the model fields with values from filter form.
 * - Execute this method to get CActiveDataProvider instance which
 * will filter
 * models according to data in model fields.      * - Pass data
 * provider to CActiveDataProvider, CActiveView or any similar widget.
 *
 * @return CActiveDataProvider the data provider that can return
 * the models
 * based on the search/filter conditions.
 */public function search()
{
// @todo Please modify the following code to remove attributes
that should not be searched.
$criteria=new CActiveCriteria;
$criteria->compare('id_dosen',$this->id_dosen);           $criteria-
>compare('nama_dosen',$this->nama_dosen,true);
$criteria->compare('nip',$this->nip,true);
$criteria->compare('tgl_lahir',$this->tgl_lahir,true);
$criteria->compare('jabatan',$this->jabatan,true);
$criteria->compare('gelar',$this->gelar,true);
$criteria->compare('pendidikan',$this-
>pendidikan,true);
$criteria->compare('bidang_keahlian',$this->bidang_keahlian,true);
$criteria->compare('pangkat',$this->pangkat,true);
return new CActiveDataProvider($this, array(
'criteria'=>$criteria,
));
}
/**
 * Returns the static model of the specified AR class.
 * Please note that you should have this exact method in all
 * your CActiveRecord descendants!
 * @param string $className active record class name.
 * @return Dosen the static model class
 */public static function model($className=__CLASS__)
{
return parent::model($className);
}
}
}

```

## **b.View admin.php**

### **(Daftar Dosen)**

```

<?php
/* @var $this DosenController */
/* @var $model Dosen */
$this->breadcrumbs=array(
'Dosens'=>array('index'),
'Manage',

```

```

);
$this->menu=array(
//array('label'=>'List Dosen', 'url'=>array('index')),
array('label'=>'Tambah Daftar',
'url'=>array('create'),'visible'=>Yii::app()->user-
>checkAccess('Staff')),
);
Yii::app()->clientScript->registerScript('search', "
$('.search button').click(function(){
$('.search-form').toggle(); return
false;
});
$('.search-form form').submit(function(){
$('#dosen-grid').yiiGridView('update', { data:
$(this).serialize()
});
return false;
});
");
?>
<h1>Daftar Dosen</h1>
<?php $this->widget('application.extensions.tablesorter.Sorter',
array(
'data'=>$records,
'columns'=>array(
'nama_dosen',
'nip',
'tgl_lahir',
'bidang_keahlian',
'jabatan',
'gelar',
/*
'pendidikan',
'pangkat',
*/
),
)); ?>

```

#### **create.php**

```

<?php
/* @var $this DosenController */
/* @var $model Dosen */
$this->breadcrumbs=array(
'Dosens'=>array('index'),
'Create',
);
$this->menu=array(
//array('label'=>'List Dosen', 'url'=>array('index')),
array('label'=>'Daftar Dosen', 'url'=>array('admin')),
);
?>
<h1>Tambah Data Dosen</h1>
<?php $this->renderPartial('_form', array('model'=>$model)); ?>

```

#### **\_form.php**

```

<?php

```

```

/* @var $this DosenController */
/* @var $model Dosen */
/* @var $form CActiveForm */
?>
<div class="form">
<?php $form=$this->beginWidget('CActiveForm', array( 'id'=>'dosen-
form',
// Please note: When you enable ajax validation, make sure the
corresponding
// controller action is handling ajax validation correctly.
// There is a call to performAjaxValidation() commented in
generated controller code.
// See class documentation of CActiveForm for details on this.
'enableAjaxValidation'=>false,
)); ?>
<p class="note">Field dengan tanda <span
class="required">*/</span> harus diisi.</p><p>
class="note">Isi NIP Tanpa Spasi</p>
<?php echo $form->errorSummary($model); ?>
<div class="row">
<?php echo $form->labelEx($model, 'nama_dosen'); ?>
<?php echo $form-
>textField($model, 'nama_dosen', array('size'=>50, 'maxlength'=>50))
;
?>
<?php echo $form->error($model, 'nama_dosen'); ?>
</div>
<div class="row">
<?php echo $form->labelEx($model, 'nip'); ?>
<?php echo $form-
>textField($model, 'nip', array('size'=>30, 'maxlength'=>30)); ?>
<?php echo $form->error($model, 'nip'); ?>
</div>
<div class="row">
<?php echo $form->labelEx($model, 'tgl_lahir'); ?>
<?php $this->widget('zii.widgets.jui.CJuiDatePicker',
array(
// you must specify name or
model/attribute
'model'=>$model,
'attribute'=>'tgl_lahir',
'value' => $model->tgl_lahir,
'options' => array(
'dateFormat'=>'yy-mm-dd',
'showOn'=>'button',
'yearRange'=>'-60',
'changeMonth' => 'true',
'changeYear' => 'true',
'constrainInput' => 'false',
'duration'=>'fast',
'showAnim' =>'slide',
// 'showAnim' =>'fold',
),
'htmlOptions'=>array('size'=>20,),
)
); ?>

```

```

<?php echo $form->error($model,'tgl_lahir'); ?>
</div>
<div class="row">
<?php echo $form->labelEx($model,'jabatan'); ?>
<?php echo $form-
>textField($model,'jabatan',array('size'=>20,'maxlength'=>20)); ?>
<?php echo $form->error($model,'jabatan'); ?>
</div>
<div class="row">
<?php echo $form->labelEx($model,'gelar'); ?>
<?php echo $form-
>textField($model,'gelar',array('size'=>20,'maxlength'=>20)); ?>
<?php echo $form->error($model,'gelar'); ?>
</div>
<div class="row">
<?php echo $form->labelEx($model,'pendidikan'); ?>
<?php echo $form-
>textField($model,'pendidikan',array('size'=>20,'maxlength'=>100))
; ?>
<?php echo $form->error($model,'pendidikan'); ?>
</div>
<div class="row">
<?php echo $form->labelEx($model,'bidang_keahlian');
?>
<?php echo $form-
>textField($model,'bidang_keahlian',array('size'=>100,'maxlength'=>
>100)); ?>
<?php echo $form->error($model,'bidang_keahlian'); ?>
</div>
<div class="row">
<?php echo $form->labelEx($model,'pangkat'); ?>
<?php echo $form-
>textField($model,'pangkat',array('size'=>30,'maxlength'=>30)); ?>
<?php echo $form->error($model,'pangkat'); ?>
</div>
<div class="row buttons">
<?php echo CHtml::submitButton($model->isNewRecord ?
'Tambah' : 'Save'); ?>
</div>
<?php $this->endWidget();
?></div><!-- form -->

```

#### **view.php**

```

<?php
/* @var $this DosenController */
/* @var $model Dosen */
$this->breadcrumbs=array(
'Dosens'=>array('index'),
$model->id_dosen,
);
$this->menu=array(
//array('label'=>'List Dosen', 'url'=>array('index')),
array('label'=>'Tambah Data',
'url'=>array('create'),'visible'=>Yii::app()->user-
>checkAccess('Staff')), array('label'=>'Ubah
Data', 'url'=>array('update',

```

```

'id'=>$model->id_dosen), 'visible'=>Yii::app()->user-
>checkAccess('Staff')),
array('label'=>'Hapus Data', 'url'=>'#',
'linkOptions'=>array('submit'=>array('delete', 'id'=>$model-
>id_dosen), 'confirm'=>'Anda yakin?'), 'visible'=>Yii::app()->user-
>checkAccess('Staff')),
    array('label'=>'Daftar Dosen', 'url'=>array('admin')),
);
?>
<h1>Detail Dosen <?php echo $model->nip; ?></h1>
<?php $this->widget('zii.widgets.CDetailView', array(
'data'=>$model,
'attributes'=>array(
'nama_dosen',
'nip',
'tgl_lahir',
'jabatan',
'gelar',
'pendidikan',
'bidang_keahlian',
'pangkat',
),
)); ?>
view.php
<?php
/* @var $this DosenController */
/* @var $data Dosen */
?>
<div class="view">
<b><?php echo CHtml::encode($data->getAttributeLabel('nip'));
?></b>
<?php echo CHtml::link(CHtml::encode($data->nip),
array('view', 'id'=>$data->id_dosen)); ?>
<br />
<b><?php echo CHtml::encode($data->getAttributeLabel('nama_dosen')); ?></b>
<?php echo CHtml::encode($data->nama_dosen); ?>
<br />
<b><?php echo CHtml::encode($data->getAttributeLabel('tgl_lahir')); ?></b><?php
echo CHtml::encode($data->tgl_lahir); ?><br />
<b><?php echo CHtml::encode($data->getAttributeLabel('bidang_keahlian')); ?></b>
<?php echo CHtml::encode($data->bidang_keahlian); ?>
<br />
<b><?php echo CHtml::encode($data->getAttributeLabel('jabatan')); ?></b>
<?php echo CHtml::encode($data->jabatan); ?>
<br />
<b><?php echo CHtml::encode($data->getAttributeLabel('gelar'));
?></b>
<?php echo CHtml::encode($data->gelar); ?>
<br />
<?php
/*

```

```

<b><?php
CHtml::encode($data>getAttributeLabel('pendidikan')); ?></b>      echo
<?php echo CHtml::encode($data->pendidikan); ?>
<br />
<b><?php echo CHtml::encode($data-
>getAttributeLabel('pangkat')); ?></b>
<?php echo CHtml::encode($data->pangkat); ?>
<br />
*/ ?>
</div>

```

### **c. Controller**

#### **DosenController.php**

```

<?php
class DosenController extends Controller
{
/**
 * @var string the default layout for the views. Defaults to
 * '//layouts/column2', meaning
 * using two-column layout. See
 * 'protected/views/layouts/column2.php'.
 */public $layout='//layouts/column2';
/**
 * @return array action filters
 */public function filters()
{
return array(
    //'rights',
    'accessControl', // perform access control for
    CRUD operations
    'postOnly + delete', // we only allow deletion via POST request
);
}

/**
 * Specifies the access control rules.
 * This method is used by the 'accessControl' filter.
 * @return array access control rules
 */public function accessRules()
{
return array(
    array('allow', // allow authenticated user to
    perform 'create' and 'update' actions
    'actions'=>array('index','view','admin'),
    'users'=>array('@'),
    ),
    array('allow', // allow admin user to perform
    'admin' and 'delete' actions

    'actions'=>array('create','update','delete'),
    'hak akses'=>array('Staff','Kaprodi','SuperAdmin'),
    ),
    array('deny', // deny all users
    'users'=>array('*'),
    ),
);
};

```

```

}

/**
 * Displays a particular model.
 * @param integer $id the ID of the model to be displayed
 */public function actionView($id)
{
$this->render('view',array(
'model'=>$this->loadModel($id),
));
}

/**
 * Creates a new model.
 * If creation is successful, the browser will be redirected to
the 'view' page.
 */public function createAction()
{
$model=new Dosen;

// Uncomment the following line if AJAX validation is needed
// $this->performAjaxValidation($model);

if(isset($_POST['Dosen']))
{
$model->attributes=$_POST['Dosen'];
if($model->save())
$this->redirect(array('view','id'=>$model-
>id_dosen));
}

$this->render('create',array(
'model'=>$model,
));
}

/**
 * Updates a particular model.
 * If update is successful, the browser will be redirected to the
'view' page.
 * @param integer $id the ID of the model to be updated
 */public function actionUpdate($id)
{
$model=$this->loadModel($id);

// Uncomment the following line if AJAX validation is needed
// $this->performAjaxValidation($model);

if(isset($_POST['Dosen']))
{
$model->attributes=$_POST['Dosen'];
if($model-
>save())
$this->redirect(array('view','id'=>$model-
>id_dosen));
}
}

```

```

$this->render('update',array(
    'model'=>$model,
));
}

/**
 * Deletes a particular model.
 * If deletion is successful, the browser will be redirected to
    the 'admin' page.
 * @param integer $id the ID of the model to be deleted
 */public function actionDelete($id)
{
    $this->loadModel($id)->delete();

    // if AJAX request (triggered by deletion via admin
    grid view), we should not redirect the browser
    if(!isset($_GET['ajax']))
    $this->redirect(isset($_POST['returnUrl']) ?
    $_POST['returnUrl'] : array('admin'));
}
/**
 * Lists all models.
 */public function actionIndex()
{
    $dataProvider=new CActiveDataProvider('Dosen');
    $this->render('index',array(
        'dataProvider'=>$dataProvider,
    ));
}

/**
 * Manages all models.
 */public function actionAdmin()
{
    $records=Dosen::model()->findAll();
    $this->render('admin',array(
        'records'=>$records,
    ));
}

/**
 * Returns the data model based on the primary key given in the
    GET variable.
 * If the data model is not found, an HTTP exception will be
    raised.
 * @param integer $id the ID of the model to be loaded
 * @return Dosen the loaded model
 * @throws CHttpException
 */public function loadModel($id)
{
    $model=Dosen::model()->findByPk($id);
    if($model===null) throw new CHttpException(404,'The requested page does not exist.');
```

```

/**
 * Performs the AJAX validation.
 * @param Dosen $model the model to be validated
 */protected function performAjaxValidation($model)
{
if(isset($_POST['ajax']) && $_POST['ajax']==='dosenform')
{
echo CActiveForm::validate($model);
Yii::app()->end();
}
}
}
}
}
}
6.Hak akses
a.Model AuthAssignment.php
<?php

/**
 * This is the model class for table "AuthAssignment".
 *
 * The followings are the available columns in table
 'AuthAssignment':
 * @property string $itemname
 * @property string $userid
 * @property string $bizrule
 * @property string $data
 *
 * The followings are the available model relations:
 */class AuthAssignment extends CActiveRecord
{
/**
 * Returns the static model of the specified AR class.
 * @return AuthAssignment the static model class
 */public static function model($className=__CLASS__)
{
return parent::model($className);
}

/**
 *
 * @desc check if User is assigned to an AuthItem
 * @param string $username from table user.username
 * @param string $itemname from table authAssignment.itemname
 * @return boolean
 */public static function userIsAssigned($userid, $itemname)
{
return (AuthAssignment::model()-
>findByAttributes(array('userid'=>$userid,
'itemname'=>$itemname))
!== null ? true : false);
}

/**
 * @return string the associated database table name
 */public function tableName()
{
return 'AuthAssignment';
}

```

```

}

/**
 * @return array validation rules for model attributes.
 */
public function
rules()
{
// NOTE: you should only define rules for those attributes that
// will receive user inputs.
return array(
array('itemname, userid', 'required'),
array('itemname, userid', 'length', 'max'=>64),
array('bizrule, data', 'safe'), // The following
rule is used by search(). // Please remove those
attributes that should not be searched.
//array('itemname, userid, bizrule, data', 'safe',
'on'=>'search'),
);
}
/**
 * @return array relational rules.
 */
public function relations()
{
return array(
'user'=>array(
self::HAS_MANY,
'User',
'username'
)
);
}
/**
 * @return array customized attribute labels (name=>label)
 */
public function attributeLabels()
{
return array(
'itemname' => 'Itemname',
'userid' => 'Username',
'bizrule' => 'BizRule',
'data' => 'Data',
);
}
}

```

### **AuthItem.php**

```

<?php
/**
 * This is the model class for table "AuthItem".
 *
 * The followings are the available columns in table 'AuthItem':
 * @property string $name
 * @property integer $type
 * @property string $description
 * @property string $bizrule
 * @property string $data
 *
 */

```

```

* The followings are the available model relations:
*/class AuthItem extends CActiveRecord
{
/**
* Returns the static model of the specified AR class.
* @return AuthItem the static model class
*/public static function model($className=__CLASS__)
{
return parent::model($className);
}
/**
* @return string the associated database table name
*/public function tableName()
{
return 'AuthItem';
}
/**
* @return array validation rules for model attributes.
*/public function rules()
{
if(isset($_GET['item'])) return array(array('name',
'safe'));
return array(
array('name, type, 'required'), array('type',
'numerical', 'integerOnly'=>true), array('name', 'length',
'max'=>64), array('description, bizrule, data',
'safe'), // The following rule is used by search().
// Please remove those attributes that should not be
searched.
array('name, type, description, bizrule, data',
'safe', 'on'=>'search'),
);
}
/**
* @return array relational rules.
*/public function relations()
{
return array(
/*
* These both are used to find AuthItem they are not bound in
AuthItemChild
* and don't appear in the RBAC Tree.
* Relation 'childs' is exactly the same as Relation 'unbound
Child'
* look in AuthItemChild::_findUnboundItems()
*/
'unboundParents' => array( // LEFT OUTER JOIN
AuthItemChild AS unboundParents ON t.name = unboundParents.parent
self::HAS_MANY,
'AuthItemChild',
'parent',
'select' => 'unboundParents.parent
AS uParent',
),
'unboundChilds' => array( // LEFT OUTER JOIN
AuthItemChild AS unboundChilds ON t.name = unboundChilds.child

```

```

                self::HAS_MANY,
                'AuthItemChild',
                'child',
                'select' => 'unboundChilds.child AS
uChild',
            ),
        );
    }

/**
 * @desc If Items are not listed in table AuthItemChild, they
 * don't appear in the RBAC Tree.
 * @desc So we have to find them seperately
 */
public function findUnboundItems()
{
    /*
    *SELECT `t`.`name` AS `t0_c0`,
    *      `t`.`type` AS `t0_c1`,
    *      `t`.`description` AS `t0_c2`,
    *      `t`.`bizrule` AS `t0_c3`,
    *      `t`.`data` AS `t0_c4`,
    *      `unboundParents`.`parent` AS `t1_c0`,
    *      `unboundParents`.`child` AS `t1_c1`,
    *      `unboundChilds`.`parent` AS `t2_c0`,
    *      `unboundChilds`.`child` AS `t2_c1`
    * FROM `AuthItem` `t`
    * LEFT OUTER JOIN `AuthItemChild` `unboundParents`
    * ON (`unboundParents`.`parent`=`t`.`name`)
    * LEFT OUTER JOIN `AuthItemChild` `unboundChilds`
    * ON (`unboundChilds`.`child`=`t`.`name`)
    *      * WHERE (unboundParents.parent is Null AND
    *              unboundChilds.child is NULL)
    * ORDER BY t.parent
    */
    $result = $this->findAll(array(
        'with' => array('unboundParents',
            'unboundChilds'),
        'condition' => 'unboundParents.parent is
Null AND unboundChilds.child is NULL',
        'order' => 't.type ASC, t.name DESC'
    )
    );
    return $result;
}
/**
 * @return array customized attribute labels (name=>label)
 */
public function attributeLabels()
{
    return array(
        'name' =>
        'Item Name',
        'type' => 'Type',
        'description' => 'Description',
        'bizrule' => 'Bizrule',
    );
}

```

```

'data' => 'Data',
);
}
/**
 *   Retrieves a list of models based on the current
search/filter conditions.
 *   @return CActiveDataProvider the data provider that can
return the models based on the search/filter conditions.
 */
public function search()
{
// Warning: Please modify the following code to remove attributes
that
// should not be searched.
$criteria=new CDbCriteria;
$criteria->compare('name',$this->name,true);
$criteria->compare('type',$this->type);
$criteria->compare('description',$this-
>description,true);
        $criteria->compare('bizrule',$this->bizrule,true);
        $criteria->compare('data',$this->data,true);

        return new CActiveDataProvider(get_class($this),
array(
                'criteria'=>$criteria,
                ));
}
}

```

#### **AuthItemChild.php**

```

<?php
/**
 * This Class manage the Access Items Tree
 * It does create, move and delete AuthItems
 *
 * The main action is to build an array Tree with AuthItems
 *
 */
class AuthItemChild extends
CActiveRecord
{
public $parent;
public $child;
/**
 * Returns the static model of the specified AR class.
 * @return AuthItemChild the static model class
 */
public static function model($className=__CLASS__)
{
return parent::model($className);
}
/**
 * @return string the associated database table name
 */
public function tableName()
{
return 'AuthItemChild';
}
}

```

```

}
/**
 * @see framework/base/CModel::rules()
 */
public function rules()
{
return array(
array('parent, child', 'required'), //
array('parent, child', 'safe'), // is a child of all other
validations
);
}
/**
 * @return array relational rules.
 */
public function relations()
{
return array(
/*
 * These both are used tu build the RBAC Tree
 * look in $this->getItemTree()
 */
'parents' => array( // LEFT OUTER JOIN
AuthItemChild AS parents ON AuthItemChild.parent = parents.child
self::HAS_MANY, 'AuthItemChild',
'child',
),
'childs' => array( // LEFT OUTER JOIN AuthItem
AS childs ON (AuthItemChild.child = childs.name)
self::BELONGS_TO,
'AuthItem',
'child',
),
);
}
}

```

### **User.php**

```

<?php
/**
 *
 * feel free to mod this Class
 *
 */
class User extends CActiveRecord
{
public static function model($className=__CLASS__)
{
return parent::model($className);
}
/**
 * @return string the associated database table name
 */
public function tableName()
{
// DO NOT CHANGE
return Yii::app()->controller->module->tableUser;
}
}

```

```

// or use your known table name like
// return 'TableUser';
}
public function rules()
{
return array();
}
public function relations()
{
return array();
}

public function attributeLabels()
{
return array();
}
}

```

## **b. View**

### **1) Assignmentsindex.php**

```

<?php
$this->breadcrumbs=array(
'Assignments'=>$this->createUrl('//rbac/assignment'),
);
?>
<h1><b>Hak akses Pengguna</b> - Userlist</h1><br>
<?php
// form errors and success $model->renderMessages();
// pager
$this->widget('CLinkPager', array(
'pages' => $pages,
))
?><br><br><?php
// Form Userlist with Hak akses
$this->renderPartial('_listForm', array('getVars'=>$getVars,
'users'=>$users, 'colUsername'=>$colUsername,
'colUserid'=>$colUserid));
/ pager
$this->widget('CLinkPager', array(
'pages' => $pages,
))
?>

```

### **manage.php**

```

<?php
$this->breadcrumbs=array(
'Assignments'=>$this->createUrl('//rbac/assignment'),
'Manage',
);
?>
<h1>Hak akses Assignments - Assign/Eject Hak akses</h1><br>
<?php
// form errors and success $model->renderMessages();
?>
<b>Assign/Eject Hak akses from:
<table>

```

```

<tr>
<th>User</th>
<th>Email</th>
</tr>
<tr>
<td><?php echo CHtml::encode($manageUser->$colUsername)?></td>
<td><?php echo CHtml::encode($manageUser->$colEmail)?></td>
</tr>
</table>
<b>Or back to: <a href="<?php echo $this->createUrl('//rbac/assignment', $getVars)?>">Userlist</a>, <a href="<?php echo $this->createUrl('//rbac/assignment/edit', array_merge(array('userid'=>$manageUser->$colUserid), $getVars)?>">edit Assignments</a></b>.
</b>
<?php if($manageUser && $displayHelper->hasItems()):
$this->renderPartial('_treeForm',
array(
'displayHelper'=>$displayHelper,
'getVars'=>$getVars,
'manageUser'=>$manageUser,
'colUsername'=>$colUsername,
'colUserid'=>$colUserid
)
);
endif;
?>
<br><b>Note:<br></b>
After a Buisness Rule the AccessController ends with the Buisness Rule Result and does not climp up to Parent any more.<br>
Check Access steps in Detail: <br>
1.) if User has requested Item Assigned do 2) - else do 4)<br>
2.) if requested User Assignment Item has Buisness Rule check and end - else do 3)<br>
3.) if requested RBAC Item has Buisness Rule check and end - else end with true<br>
4.) if requested Item has parent do 1) with parent as requested Item - else end with false<br><br>
The Data field is not used while checking access. Feel free to use it fore your needs.

```

### **\_listForm.php**

```

<div>
<?php echo CHtml::beginForm($this->createUrl('//rbac/assignment'), 'get') ?>
<table style="width:70%">
<tr>
<th><?php echo CHtml::textField('s1', $getVars['s1'], array('size'=>'10', 'onChange'=>'this.form.submit()')) ?></th>
<th><?php echo CHtml::textField('s2', $getVars['s2'], array('size'=>'10', 'onChange'=>'this.form.submit()')) ?></th>
<th><?php echo CHtml::textField('s3',

```

```

$getVar['s3'], array('size'=>'15',
'onChange'=>'this.form.submit()')) ?></th>
<th><?php echo CHtml::textField('s4',
$getVar['s4'], array('size'=>'10',
'onChange'=>'this.form.submit()')) ?></th>
</tr>
<?php
$lastUser='';
$style='style="border-top:1px solid #cdcdcd"'; foreach($users
as $user):
if($user[$coluserid] == $lastUser) $style='';
?>
<tr>
<td <?php echo $style ?><?php echo ($lastUser !=
$user[$coluserid]
?
'<a
href="'. $this->createUrl('//rbac/assignment/manage',
array_merge(array('userid'=>$user[$coluserid]), $getVar)).'">'.CHT
ml::encode($user[$colusername]).'</a>' : '&nbsp;') ?></td> <td
<?php echo $style ?><?php echo '<a href="'. $this-
>createUrl('//rbac/assignment/edit',
array_merge(array('userid'=>$user[$coluserid]), $getVar)).'">'.CHT
ml::encode($user['itemname']).'</a>' ?></td> <td <?php echo
$style
?> nowrap
><?php echo
nl2br(CHtml::encode($user['bizrule'])) ?></td>
<td <?php echo $style ?><?php echo
CHtml::encode($user['data']) ?></td>
</tr>
<?php
$lastUser=$user[$coluserid];
$style='style="border-top:1px solid #cdcdcd"';
endforeach; ?>
</table>
<?php echo CHtml::endForm() ?>
</div>

```

### edit.php

```

<?php
$this->breadcrumbs=array(
'Assignments'=>$this->createUrl('//rbac/assignment'),
'Edit',
);
?>
<h1>Edit Hak akses Assignmments</h1><br>
<?php
// form errors and success
$model->renderMessages();
?>
<h3>from:</h3>
<table>
<tr>
<th>Username</th>
<th>Email</th>
</tr>
<tr>
<td><?php echo CHtml::encode($user->$colUsername) ?>
<td><?php echo CHtml::encode($user->$colEmail) ?>

```

```

</td>
</table>
<b>Or back to: <a href="<?php echo $this-
>createUrl('//rbac/assignment', $getVars)?">Userlist</a>,
<a href="<?php echo $this->createUrl('//rbac/assignment/manage',
array_merge(array('userid'=>$user-
>$colUserId), $getVars))?">Assign/Eject Hak akses</a></b><br><br>
<?php if(count($assignments)):
    $this->renderPartial('_editForm', array('getVars'=>$getVars,
'assignments'=>$assignments, 'user'=>$user,
'colUsername'=>$colUsername, 'colUserId'=>$colUserId));
else:
?>User has no <a href="<?php echo
$this->createUrl('//rbac/assignment/manage',
array_merge(array('userid'=>$user-
>$colUserId), $getVars))?">Assignments</a></b><br><br>
<?php
endi;

```

### **\_editForm.php**

```

<div>
<?php echo CHtml::beginForm($this-
>createUrl('//rbac/assignment/edit', $getVars), 'post') ?>
<?php echo CHtml::hiddenField('username', $user->$colUsername) ?>
<?php echo CHtml::hiddenField('userid', $user->$colUserId) ?>
<table>
<tr>
<th>Hak akses</th>
<th>Buisness Rule</th>
<th>Data</th>
</tr>
<?php foreach($assignments as $assignment): ?>
<tr>
<td><b><?php echo CHtml::encode($assignment-
>itemName) ?></b></td>
<td><?php echo
CHtml::textArea("assignments[".$assignment->itemName."][bizrule]",
$assignment->bizrule, array('cols'=>60, 'rows'=>5)) ?></td>
<td><?php echo
CHtml::textArea("assignments[".$assignment->itemName."][data]",
$assignment->data, array('cols'=>30, 'rows'=>5)) ?></td>
</tr>
<?php endforeach ?>
</table>
<span style="padding-left:4em"><?php echo
CHtml::submitButton('Save Changes') ?></span>
<?php echo CHtml::endForm() ?>
</div>

```

### **2)RBAC Tree index.php**

```

<?php
$this->breadcrumbs=array(
'Manage Rbac Tree'=>$this->createUrl('//rbac/rbac'),
);
?>
<h1>Daftar Hak akses</h1>

```

```

<?php
// form errors and success
$model->renderMessages();
// edit form if($editItem !== null || !$displayHelper-
>hasItems()):    if(!$displayHelper->hasItems()):
    $editItem=new AuthItem;    endif;
$this->renderPartial('_editForm',
array('editItem'=>$editItem, 'displayHelper'=>$displayHelper,
'nameFilter'=>$nameFilter)); endif;
// eject form if(count($ejectItemsGet)):
$this->renderPartial('_ejectForm',
array('ejectItemsGet'=>$ejectItemsGet,
'displayHelper'=>$displayHelper)); endif;
// edit form if($treeFormat ==
'renderItemTree'):
?>
<br>
<a href="<?php echo $this-
>createUrl('//rbac/rbac',array('createNew'=>1))    ?>">Tambah    Hak
akses Baru</a>
<br>
<?php
endif; if($displayHelper->hasItems()):
$this->renderPartial('_treeForm',
array('displayHelper'=>$displayHelper,
'treeFormat'=>$treeFormat)); endif;
?>
<br><b>Note:<br></b>
After a Buisness Rule the AccessController ends with the Buisness
Rule Result and does not climb up to Parent any more.<br>
Check Access steps in Detail: <br>
1.) if User has requested Item Assigned do 2) - else do 4)<br> 2.)
if requested User Assignment Item has Buisness Rule check and end
- else do 3)<br>
3.) if requested RBAC Item has Buisness Rule check and end -
else end with true<br>
4.) if requested Item has parent do 1) with parent as requested
Item - else end with false<br><br>
The Data field is not used while checking access. Feel free to use
it fore your needs.

```

#### treeForm.php

```

<?php
if($treeFormat == 'renderItemTree'):
?>
<div>
<?php echo CHtml::beginForm($this>createUrl('//rbac/rbac/move'),
'post') ?>
<br>
<?php echo CHtml::submitButton('Move Item') ?>
<div style="width: 90%">
<?php echo $displayHelper->renderTree(); ?>
</div>
<br>
<?php echo CHtml::submitButton('Move Item') ?>
<?php echo CHtml::endForm() ?>

```

```

</div>
<br>
<a href="<?php echo $this->createUrl('//rbac/rbac',array('generateCode'=>1)) ?>">Generate
PHP Code</a>
<br>
<?php elseif($treeFormat ==
'renderCode'):
?>
<br> <?php
echo $displayHelper->renderTree("\$auth=Yii::app()->authManager;\n");
?>
<br>
<a href="<?php echo $this->createUrl('//rbac/rbac')
?>">Return to Tree View</a>
<br>
<?php elseif($treeFormat ==
'renderItemInfo'):
?>
<br> <?php
echo $displayHelper->renderTree();
?>
<?php endif;

```

#### editForm.php

```

<div>
<script type="text/javascript">
confirmDelete
= function(_t){
if(confirm("Really delete
Item?")){
_t.form.submit();
}
}
</script>
<?php echo CHtml::errorSummary($editItem, '<div
id="rbacError">', '</div>'); ?>
<?php echo CHtml::beginForm($this->createUrl('rbac/edit'), 'post')
?>
<?php echo CHtml::hiddenField('oldName', $editItem->name);
?>
<table style="width: 50%">
<tr>
<td>Item Name:</td>
<td>
<?php echo CHtml::textField
('editItem[name]', $editItem->name); ?>
</td>
<td><span style="color:#cc0000">Required (min.2
Chars)</span><br>Filter: <?php echo $nameFilter ?></td>
</tr>
<tr>
<td>Type:</td>
<td colspan="2">
<?php echo CHtml::dropDownList
('editItem[type]', $editItem->type,
array('Operation', 'Task', 'Hak akses')) ?><br/>

```

```

</td>
</tr>
<tr>
<td>Description:</td>
<td colspan="2">
<?php echo CHtml::textArea
('editItem[description]', $editItem->description,
array('cols' => 70, 'rows' => 4)) ?><br/>
</td>
</tr>
<tr>
<td>Bizrule:</td>
<td colspan="2">
<?php echo CHtml::textArea
('editItem[bizrule]', $editItem->bizrule,
array('cols' => 70, 'rows' => 4)) ?><br/>
</td>
</tr>
<tr>
<td>Data:</td>
<td colspan="2">
<?php echo CHtml::textArea
('editItem[data]', $editItem->data,
array('cols' => 70, 'rows' => 4)); ?><br/>
</td>
</tr>
<tr>
<td colspan="3">
<?php
if(!$displayHelper->hasItems() || isset($_GET['createNew'])):
echo CHtml::submitButton('Tambah Hak
akses', array('name'=>'createItem'));
else:
?>Edit/Create/Delete:<br/>
<table>
<tr>
<td>
<?php echo
CHtml::submitButton('Update Hak akses',
array('name'=>'updateItem'))
?>
</td>
<td>
<?php echo
CHtml::submitButton('Simpan Sebagai Hak akses Baru',
array('name'=>'createItem')) ?>
</td>
<td>
<?php echo
CHtml::submitButton('Hapus Hak akses '.$editItem->name.'',
array('name'=>'deleteItem',
'onClick'=>"confirmDelete(this)")) ?>
</td>
</tr>
</table>
<b>or <a href="<?php echo $this->

```

```

>createUrl('//rbac/rbac') ?>">close</a> input Fields.</b>
<?php endif ?>
</td>
</tr>
</table>
<?php echo CHtml::endForm(); ?>
</div>

```

### **ejectForm.php**

```

<?php ?>
<div style="padding: 2em; clear: both; border-left: 2px solid
#cccccc; border-top: 2px solid #cccccc; border-bottom: 2px solid
#cccccc" >
    <?php echo CHtml::beginForm($this->createUrl('rbac/eject'),
'post') ?>
    <?php echo CHtml::hiddenField ('eject[parent]',
$ejectItemsGet['parent']->name); ?>
    <?php echo CHtml::hiddenField ('eject[child]',
$ejectItemsGet['child']->name); ?>
    <h3>Eject Child <i><?php echo $ejectItemsGet['child']>name ?></i>
from Parent <i><?php echo $ejectItemsGet['parent']-
>name ?></i></h3>
    <?php
echo $displayHelper->treeFormat['groupIn'];
echo $displayHelper-
>renderItemEject($ejectItemsGet['parent']);
echo $displayHelper->treeFormat['groupIn'];
echo $displayHelper-
>renderItemEject($ejectItemsGet['child']);
?>
    <br />
    <?php echo CHtml::submitButton                ('Eject Item');
    ?>
    <?php
echo          $displayHelper->treeFormat['groupOut'];          echo
$displayHelper->treeFormat['groupOut'];
?>
    <?php echo CHtml::endForm()
    ?>
</div>

```

## **c. Controller**

### **DefaultController.php**

```

<?php
class DefaultController extends RBACBaseController
{
public function actionIndex()
{
$this->redirect(array('//rbac/rbac'));
}
}

```

### **AssignmentController.php**

```

<?php
class AssignmentController extends RBACBaseController
{
// user List
public $users;

```

```

public $authAssignments; // yii crashes by name $assignments in
CWebApplication in Line 391: if(!classExists(...)
// manage user Assignments
public $manageUser;
//     edit     Assignments     from     User
public $editUser;
public $assignments;
//CPagination     public
$paginate;
public $usersPerPage=20;
// search     public
$searchFields;
public $defaultSearchFields=array(
's1'=>'Username', // displayed default value     's2'=>'Hak akses',
's3'=>'Buisness Rule',
's4'=>'Data'
);
/**
 * Lists all models.
 */
public function actionIndex()
{
$this->checkAccess('RbacAssignmentViewer',     true);     $this->
doRender('index', array(
'assignUser'=>$this->manageUser,
'users' => $this->_getUserlist(),
'pages' => $this->paginate,
'getVars' => $this->getGetVars()
));
Yii::app()->end();
}
/**
 *
 * @desc
 */
private function _getUserlist()
{
/**
 * get changeable table- and columnnames
 */
$module=Yii::app()->controller->module;
$tblUser=$module->tableUser;
$colUsername=$module->columnUsername;
$columnUserId=$module->columnUserId;
/*
 * this is a very ugly yii workaround
 * of Paginate Tables with HAS_MANY Joins and
Conditions
 *
 * I do this with two selects:
 * one for count, limit and offset
 * and one for the Item select
 * In both we have to use the conditions
 *
 */
}

```

```

// create criteria for count, limit and offset select
$criteria=new CDbCriteria(
array(
'with'=>array('assignment'),
'order'=>'t.$colUsername ASC'
)
);
// get Getvars and diff $this->searchFields and
$this->defaultSearchFields
$this->_getSearchFields();
// combine getvars with tablenamees
$criteria=array(
's1'=>'t.$colUsername',
's2'=>'assignment.itemname',
's3'=>'assignment.bizrule',
's4'=>'assignment.data');
// array for second itemselect
$criteria=array();
// get the conditions from getvars s1, s2, s3 and s4
if(isset($this->searchFields['s1']) && $this->searchFields['s1'] !=
$this->defaultSearchFields['s1'])
{
$criteria->addSearchCondition($criteria['s1'],
$this->searchFields['s1'], true, 'AND', 'LIKE');
$criteria['s1']="AuthAssignment.$colUsername LIKE
:s1";
}
if(isset($this->searchFields['s2']) && $this->searchFields['s2']
!= $this->defaultSearchFields['s2'])
{
$criteria->addSearchCondition($criteria['s2'],
$this->searchFields['s2'], true, 'AND', 'LIKE');
$criteria['s2']="AuthAssignment.itemname LIKE
:s2";
}
if(isset($this->searchFields['s3']) && $this->searchFields['s3']
!= $this->defaultSearchFields['s3'])
{
$criteria->addSearchCondition($criteria['s3'],
$this->searchFields['s3'], true, 'AND', 'LIKE');
$criteria['s3']="AuthAssignment.bizrule LIKE
:s3";
}
if(isset($this->searchFields['s4']) && $this->searchFields['s4']
!= $this->defaultSearchFields['s4'])
{
$criteria->addSearchCondition($criteria['s4'],
$this->searchFields['s4'], true, 'AND', 'LIKE');
$criteria['s4']="AuthAssignment.data LIKE
:s4";
}
// get the DB for the count select
$db=Yii::app()->getDb();
// the first half query
$sql="SELECT COUNT(*)
FROM $tblUser

```

```

LEFT JOIN AuthAssignment
ON AuthAssignment.userid =
$tblUser.$colUsername
";
// add the WHERE condition          if(count($conditions)){
$sql.=" WHERE (" .implode(' AND ',
$conditions).") ";
}
// create the command
$command=$db->createCommand($sql);
// bind the values
foreach($conditions as $key => $value)
{
    $command->bindValue($key, '%' . $this->searchFields[$key] . '%');
}
// and finally get the REAL count of Items
$count=$command->queryScalar();
// create the pagination
$this->paginate=new CPagination($count);
$this->paginate->pageSize=$this->usersPerPage;
// after this we will find limit and offset in the criteria
$this->paginate->applyLimit($criteria);
// now begin the second select for the Items
$sql=" SELECT $tblUser.$columnUserId,
$tblUser.$colUsername, AuthAssignment.itemname,
AuthAssignment.bizrule, AuthAssignment.data FROM $tblUser LEFT
JOIN AuthAssignment ON AuthAssignment.userid =
$tblUser.$columnUserId";
// add the WHERE condition
if(count($conditions)){ $sql.=" WHERE (" .implode('
AND ', $conditions).") ";
}
// add ORDER
$sql.=" ORDER BY $tblUser.$colUsername ASC,
AuthAssignment.itemname";
// add limit and offset from $criteria
$sql.=" LIMIT " . $criteria->offset . ', ' . $criteria->limit;
// create the command
$command=$db->createCommand($sql);
// bind the values
foreach($conditions as $key => $value)
{
    $command->bindValue($key, '%' . $this->
    searchFields[$key] . '%');
}
// and here the are
$this->users=$command->queryAll();
return $this->users;
}
/**
 *
 * @desc
 */private function_getSearchFields()
{
    if($this->searchFields !== null) return;
    if(isset($_GET['s1']))

```

```

{
$resource = $_GET;
}elseif(isset($_POST['search'])){
$resource = $_POST['search'];
}else{
$this->searchFields=$this->defaultSearchFields;
return;
}
foreach($this->defaultSearchFields as $k => $v)
{
if(!isset($resource[$k]))
{
$this->searchFields[$k]='';
}else{
$this->searchFields[$k]=($resource[$k] !=
'' ? urldecode($resource[$k]) : $this->defaultSearchFields[$k]);
}
}
}
/**
 * @desc merge getvars from pagination and searchfields
 */
private function getGetVars()
{
return
                isset($_GET['page'])                ?
array_merge(array('page'=>$_GET['page']), $this->searchFields) :
$this->searchFields;
}
/**
 * @desc addassignments
 */
public function actionManage(){
// get changable collumnnames
$colUsername=Yii::app()->controller->module->columnUsername;
$colUserId=Yii::app()->controller->module>columnUserId;
// check access to view
$this->checkAccess('RbacAssignmentViewer',                true);
if(isset($_GET['userid']))
{
// warn if user is protected if(in_array($_GET['userid'], $this->protectedUsers))
$this->messageWarnings[]="Warning!    User    is    protected    by
Controller";
// user must exist
if($user=User::model()-
>findByAttributes(array("$colUserId"=>urldecode($_GET['userid'])))
){
$this->manageUser=$user;
}else{
throw new CHttpException("Selected User
".$_urldecode($_GET['username'])." does not exist");
}
}elseif(isset($_POST['userid']))
{

```

```

// check access for edit assignments                                $this-
>checkAccess('RbacAssignmentEditor', true);
if(in_array($_POST['userid'], $this-
>protectedUsers))
{
$this->messageErrors[]="Sorry, User is protected by Controller";
$this->actionIndex();
}
$username = $_POST['username'];
$userid=(int)$_POST['userid'];
if(!$user=User::model()-
>findByAttributes(array("$coluserid"=>$userid)))
throw new CHttpException("Managed User $username does not exist");
// add selected assignments
if(isset($_POST['addAssignments']))
{
// fill bizRule with deny-always code if selected from user
$bizRule=isset($_POST['secureMode']) ?
'return false;' : '';
foreach($_POST['addAssignments'] as
$itemname)
{
// add default code to bizRule if
selected
if(isset($_POST['addData'])){
$item=AuthItem::model()-
>findByAttributes(array('name'=>$itemname));
$bizRule.=$item->data;
}
// add assignment
$assignment=new AuthAssignment;
$assignment->attributes=(array('userid'=>$userid,
'itemname'=>$itemname,
'bizrule'=>$bizRule, 'data'=>''));
if(!$assignment->validate())
throw new CHttpException("New Assignment validation Error");
$assignment->save();
$this->messageSuccess[]="Assignment
$itemname succesfull added.";
}
}
// remove selected assignments
if(isset($_POST['removeAssignments']))
{
foreach($_POST['removeAssignments'] as $itemname)
{
$assignment=AuthAssignment::model()-
>findByAttributes(array('userid'=>$userid,
'itemname'=>$itemname));
$assignment->delete();
$this->messageSuccess[]="Assignment      $itemname      succesfull
removed.";
}
}
$this->manageUser=$user;
}else{

```

```

$this->actionIndex();
}
$this->manageUser=$user;
$this->_getSearchFields();
$displayHelper=new RBACDisplayHelper($this, 'renderItemAssign');
$displayHelper->setUser($this->manageUser);
$this->doRender('manage', array(
    'displayHelper'=>$displayHelper,
    'manageUser'=>$this->manageUser,
    'getVars' => $this->getGetVars()
));
Yii::app()->end();
}
/**
 * @desc eject assignment or edit assignment data
 */
public function actionEdit()
{
    $colUsername=Yii::app()->controller->module->columnUsername;
    $colUserid=Yii::app()->controller->module->columnUserid;
    $this->checkAccess('RbacAssignmentViewer', true);
    if(isset($_GET['userid']))
    {
        if(in_array($_GET['userid'], $this->protectedUsers))
        {
            $this->messageWarnings[]="Warning! User is protected by Controller";
        }
        if($this->editUser=User::model()->findByAttributes(array($colUserid=>$_GET['userid']))
        {
            $this->assignments=AuthAssignment::model()->findAllByAttributes(array('userid'=>$this->editUser->$colUserid));
            $this->_getSearchFields();
            $this->doRender('edit', array(
                'user'=>$this->editUser,
                'assignments'=>$this->assignments,
                'getVars' => $this->getGetVars()
            ));
        }else{
            throw new CHttpException("Selected User ".CHtml::encode($_GET['username'])." does not exist");
        }
    }elseif(isset($_POST['userid'])
    && isset($_POST['assignments'])
    && is_array($_POST['assignments']))
    {
        $this->checkAccess('RbacAssignmentEditor', true);
        if(in_array($_POST['userid'], $this->protectedUsers))
        {
            $this->messageErrors[]="Sorry, User is protected by Controller";
        }
    }
    $this->actionIndex();
}

```

```

}
if($this->editUser=User::model()-
>findByAttributes(array($colUserid=>$_POST['userid']))
{
foreach($_POST['assignments'] as $itemName => $values)
$modelAssign=AuthAssignment::model()-
>findByAttributes(array('itemName'=>$itemName,      'userid'=>$this-
>editUser->$colUserid));
$modelAssign->attributes=array('bizrule'=>$values['bizrule'],
'data'=>$values['data']);
if($modelAssign->validate())
{
$modelAssign->save();
$this-
>messageSuccess[]="Assignment $itemName successfull updated.
        }
}
$this->_getSearchFields();
$this->doRender('edit', array('user'=>$this->editUser,
'assignments'=>$this->assignments=AuthAssignment::model()-
>findAllByAttributes(array('userid'=>$this->editUser-
>$colUserid)),
'getVars' => $this->getGetVars()
));
Yii::app()->end();
}else{
throw new CHttpException("User
.CHtml::encode($_POST['username'])." does not exist");
}
}else{
throw new CHttpException("Not enough Data for
Edit Assignments found");
}
}
} ?>

```

### **RbacController.php**

```

<?php
class RbacController extends RBACBaseController
{
public $rejectItemsGet=array();
public $rejectItemsPost=array();
public          $editItem;
public $authItem;
public $filterNames='/^[a-zA-Z]+[a-zA-Z0-9- _]+/';
/*
* @desc this is called by default and after each action of this
controller
*/
public function actionIndex()
{
$this->checkAccess('RbacViewer', true);
$model=new RBACTree;          if(isset($_GET['createNew']))
$this->editItem=AuthItem::model();          $treeFormat      =
'renderItemTree';
if(isset($_GET['generateCode']))
{

```

```

$treeFormat = 'renderCode';
}elseif(isset($_GET['item']) ||
isset($_GET['createNew']) || (!empty($_POST) && count($this-
>messageErrors))){
$treeFormat = 'renderItemInfo';
}
$this->doRender('index', array(
'model'=>$this,
'displayHelper'=>new RBACDisplayHelper($this,
$treeFormat),
'treeFormat'=>$treeFormat,
'ejectItemsGet'=>$this->ejectItemsGet,
'ejectItemsPost'=>$this->ejectItemsPost,
'editItem'=>$this->editItem,
'authItem'=>$this->authItem,
'nameFilter'=>$this->filterNames
)
);
Yii::app()->end();
}
/**
 *
 * @desc
 */
public function actionEdit()
{
$this->checkAccess('RbacViewer', true);
$model=new AuthItem;          if(empty($_POST)){
if(isset($_GET['item']))
{
if(in_array($_GET['item'], $this->protectedItems))
{
$this->messageErrors[]="Warning!
Item is protected by Controller";
}
$model->attributes=$_GET;          if($model-
>validate()){
$name=urldecode($_GET['item']);
if($item=$model->findByAttributes(array('name'=>$name)){
// display edit Item box
$this->editItem=$item;
$this->actionIndex();
}else{
$this->messageErrors[]="The
Item you want to edit does not exist";
}else{
$this->messageErrors[]="Unsecure
Data detected. Please mail the Siteadmin if this Problem
returns.";
}
}else{
//ignore missing item and display index
$this->actionIndex();
}
}else{
$this->checkAccess('RbacEditor', true);

```

```

// filter names
$_POST['editItem']['name']=$this->filterNames;
>filterString($_POST['editItem']['name'], $this->filterNames);
$model->attributes=$_POST['editItem'];
$oldName=$_POST['oldName'];           if(in_array($oldName,
$this->protectedItems) ||
in_array($_POST['editItem']['name'], $this->protectedItems))
{
$this->messageErrors[]="Sorry, Item is
protected by Controller";
$this->actionIndex();
}
if($model->validate()){
if(isset($_POST['updateItem']))
{
$this->updateItem($_POST['editItem'], $oldName);
}elseif(isset($_POST['createItem']))
{
if(!AuthItem::model()-
>findByAttributes(array('name'=>$_POST['editItem']['name'])))
{
$model->setIsNewRecord(true);
$model->save();
$this->messageSuccess[]="Item
{$_POST['editItem']['name']} successfull created.";
}else{
$this->messageErrors[]="Create
Error:  New Item <i>{$_POST['editItem']['name']}</i>  already
exists";
$this->editItem=$model;
$this->actionIndex();
}
}elseif(isset($_POST['deleteItem']))
{
AuthItem::model()-
>deleteAllByAttributes(array('name'=>$oldName));
AuthItemChild::model()-
>deleteAllByAttributes(array('parent'=>$oldName));
AuthItemChild::model()-
>deleteAllByAttributes(array('child'=>$oldName));
AuthAssignment::model()-
>deleteAllByAttributes(array('itemname'=>$oldName));
$this->messageSuccess[]="Item
$oldName successfull deleted.";
}else{
// ignore not existing submit option and render page
}
$this->actionIndex();
}else{
//use Yii error system
$model->setIsNewRecord(true);
$this->editItem=$model;
$this->actionIndex();
}
}
}
}

```

```

/**
 *
 * @desc
 * @param unknown_type $model
 * @param unknown_type $attributes
 * @param unknown_type $oldName
 */
private function _updateItem($attributes, $oldName)
{
    if(in_array($oldName, $this->protectedItems) ||
    in_array($attributes['name'], $this->protectedItems))
    {
        $this->messageErrors[]="Sorry, Item is protected by Controller";
        $this->actionIndex();
    }
    if(!$item=AuthItem::model()-
    >findByAttributes(array('name' => $oldName)))
    {
        $this->messageErrors[]="Edit Error: Update Item does not exist";
        $this->actionIndex();
    }
    if($attributes['type']==0 && $item->type>0)
    {
        if(count(AuthItemChild::model()-
        >findAllByAttributes(array('parent'=>$oldName))))
        {
            $this->messageErrors[]="Type <i>Action</i>
            can't have Childs.<br/>Please eject Childs from <i>$oldName</i>
            before switch type to <i>Operation</i>";
            $this->editItem=$item;
            $this->actionIndex();
        }
    }
    if($attributes['name'] != $oldName)
    {
        if(AuthItem::model()-
        >findByAttributes(array('name'=>$attributes['name'])))
        {
            $this->messageErrors[]="Create Error: New
            Item <i>{$_POST['editItem']['name']}</i> already exists";
            // return;
        }
        $item->attributes=$attributes;
        $item->save();
        // update RBAC-Tree AuthItemChild bindings in parent
        $newName=$attributes['name'];
        $treeItems=AuthItemChild::model()-
        >findAllByAttributes(array('parent'=>$oldName));
        foreach($treeItems as $treeItem)
        {
            $treeItem->parent=$newName;
            $treeItem->save();
        }
        // update RBAC-Tree AuthItemChild bindings in child
        $treeItems=AuthItemChild::model()-
        >findAllByAttributes(array('child'=>$oldName));
    }
}

```

```

foreach($treeItems as $treeItem)
{
$treeItem->child=$newName;
$treeItem->save();
}
// update AuthAssignment bindings in itemname
$assignments=AuthAssignment::model()-
>findAllByAttributes(array('itemname'=>$oldName));
foreach($assignments as $assignment)
{
$assignment->itemname=$newName;
$assignment->save();
}
}else{
// simple update if primary key is same
$item->attributes=$attributes;
$item->save();
}
$this->messageSuccess[]="Item ". (!isset($newName) ?
$oldName : $newName)." successfull updated.";
}
/**
 * @desc collect _POST and _GET Data, compare and
 * validate>required them.
 * @desc on fail abort eject and actionIndex
 * @return array ('parent'=>string, 'child'=>string)
 */ private function validateEject()
{
// collect vars
if(empty($_POST))
{
if(!isset($_GET['parent'])) {
$this->messageErrors[]="Eject Error: Item
".$_GET['child']." is already ejected or in top Level";
$this->actionIndex();
}
$itemNames=array('parent'=>urldecode($_GET['parent']),
'child'=>urldecode($_GET['child']));
}else{
$itemNames=array('parent'=>urldecode($_POST['eject']['parent
']), 'child'=>urldecode($_POST['eject']['child']));
}
$model=new AuthItemChild;
$model->attributes=$itemNames;
if(!$model->validate())
{
$this->messageErrors[]="Unsecure Data detected.
Please mail the Siteadmin if this Problem returns.";
$this->actionIndex();
}
return
$itemNames;
}
/**
 *
 */ public function actionEject()

```

```

{
$this->checkAccess('RbacViewer', true);
$itemNames=$this->validateEject();
$model=new AuthItem();
// check only if parent is protected
if(in_array($itemNames['parent'], $this->protectedItems))
{
if(in_array($itemNames['child'], $this->protectedItems))
{
$this->messageErrors[]="Warning! Item is protected by Controller";
}
}
}
$child=$model->findByAttributes(array('name'=>$itemNames['child']));
$parent=$model->findByAttributes(array('name'=>$itemNames['parent']));
if(empty($_POST)){
if(!$child || !$parent){
$this->messageErrors[]="One ore more
Item(s) does not exist";
$this->actionIndex();
}else{
$this->ejectItemsGet=array('parent'=>$parent, 'child'=>$child);
}
}else{
$this->checkAccess('RbacEditor', true);
$model=AuthItemChild::model();
$model->attributes=array('parent'=>$parent->name, 'child'=>$child->name);
$model->delete();
$this->messageSuccess[]="Item {$child->name}
successfull ejected from {$parent->name}.";
}
$this->actionIndex();
}
/**
 *
 * @desc
 */ public function actionMove()
{
$this->checkAccess('RbacViewer', true);
if(!empty($_POST))
{
$this->checkAccess('RbacEditor', true);
$from=isset($_POST['moveFromItem']) ?
$_POST['moveFromItem'] : null;
$to=isset($_POST['moveToItem']) ?
$_POST['moveToItem'] : null;

// check only if parent is protected
if(in_array($to, $this->protectedItems))
{
if(in_array($from, $this->protectedItems))
{

```

```

$this->messageErrors[]="Sorry, Item
is protected by Controller";
$this->actionIndex();
}
}
if(!$from || !$to || $from==$to)
{
$this->messageErrors[]="Please select
Parent- and Childitem and care that they are not same.";
$this->actionIndex();
}
// default validate
$model=new AuthItemChild;
$model->attributes=array('child'=>$from,
'parent'=>$to);
if(!$model->validate())
{
$this->messageErrors[]="Post validation
Error. Please mail Siteadmin if this Error returns.";
$this->actionIndex();
}
// check if branch already exists
if($model-
>findByAttributes(array('child'=>$from, 'parent'=>$to)) !== null)
{
$this->messageErrors[]="Create Branch
Error: Branch already exists.";
$this->actionIndex();
}
// Items exist?
$model=new AuthItem;
if(!count($model-
>findByAttributes(array('name'=>$from))) || !count($model-
>findByAttributes(array('name'=>$to))))
{
$this->messageErrors[]="Check Items exists
Error. Please mail Siteadmin if this Error returns.";
$this->actionIndex();
}
// make recursioncheck and move Items
$model=new RBACTree;
$model->moveFrom=$from;
$model->moveTo=$to;
if($model->checkRecursion())
{
$model->moveItem();
$this->messageSuccess[]="Item $from successfull moved to $to.";
}else{
$this->messageErrors[]="Can't move Selection cause that will
produce a Recursion.
<br>If you can't see producing a Recursion, it may help to eject
Item before moving it.";
$this->actionIndex();
}
}
}
$this->actionIndex();

```

```
}  
}
```

### 7. Pesan a. Model Mailbox.php

```
<?php  
/**  
 * This is the model class for table "{{mailbox_conversation}}".  
 *  
 * The followings are the available columns in table  
 * "{{mailbox_conversation}}":  
 * @property string $conversation_id  
 * @property integer $initiator_id  
 * @property integer $interlocutor_id  
 * @property string $subject  
 * @property integer $bm_read  
 * @property integer $bm_deleted  
 * @property string $modified  
 * @property string $is_system  
 */  
class Mailbox extends CActiveRecord  
{  
    const INITIATOR_FLAG = 1;  
    const INTERLOCUTOR_FLAG = 2;  
    public $to;  
    public $text;  
    public $sender_id; public  
    $is_replied;  
    /**  
     * Returns the static model of the specified AR class.  
     * @param string $className active record class name.  
     * @return Mailbox the static model class  
     */ public static function model($className=__CLASS__)  
    {  
        return parent::model($className);  
    }  
    /**  
     * @return string the associated database table name  
     */ public function tableName()  
    {  
        return MailboxModule::TBL_CONV;  
    }  
    /**  
     * @return array validation rules for model attributes.  
     */ public function rules()  
    {  
        // NOTE: you should only define rules for those attributes that  
        // will receive user inputs.  
        return array(  
            //array('initiator_del,interlocutor_del',  
            'default', 'setOnEmpty'=>true, 'value'=>null ),  
            array('initiator_id, interlocutor_id, modified',  
            'required'),  
            array('initiator_id, interlocutor_id, bm_read,  
            bm_deleted, initiator_del, interlocutor_del', 'numerical',  
            'integerOnly'=>true),  
        );  
    }  
}
```

```

array('subject', 'length', 'max'=>100), array('modified',
'length', 'max'=>10), array('is_system', 'length', 'max'=>3),
array('subject','match','pattern'=>' /^[^'.Yii::app()-
>controller->module->allowableCharsSubject.']+/i','not'=>>true),
// The following rule is used by search().
// Please remove those attributes that should not be searched.
array('conversation_id, initiator_id, interlocutor_id, subject,
bm_read, bm_deleted, modified, is_system', 'safe',
'on'=>'search'),
array('to','length','max'=>30),);
}
/**
 * @return array relational rules.
 */ public function relations()
{
// NOTE: you may need to adjust the relation name and the related
// class name for the relations automatically generated below.
return array(
'initiator'=>array(self::BELONGS_TO,'User','initiator_id'),
'interlocutor'=>array(self::BELONGS_TO,'User','interlocutor_id'),
'messages'=>array(self::HAS_MANY,'Message','conversation_id'),
);
}
/**
 * @return array customized attribute labels (name=>label)
 */ public function attributeLabels()
{
return array(
'conversation_id' => 'Id',
'initiator_id' => 'From Id',
'interlocutor_id' => 'To Id',
'to' => 'To',
'subject' => 'Subject',
'last_message_ts' => 'Last Message Received',
'modified' => 'Last Modified',
'is_system' => 'Is System',
);
}
/**
 * Retrieves a list of models based on the current
search/filter conditions.
 * @return CActiveDataProvider the data provider that can
return the models based on the search/filter conditions.
 */ public function search()
{
// Warning: Please modify the following code to remove attributes
that
// should not be searched.
$criteria=new CDbCriteria;
$criteria->compare('conversation_id',$this->conversation_id,true);
$criteria->compare('initiator_id',$this->initiator_id);
$criteria->compare('interlocutor_id',$this->interlocutor_id);
$criteria->compare('subject',$this->subject,true);
$criteria->compare('bm_read',$this->bm_read); $criteria-
>compare('bm_deleted',$this->bm_deleted);

```

```

$criteria->compare('modified',$this->modified,true);    $criteria-
>compare('is_system',$this->is_system,true);
return new CActiveDataProvider($this, array(
'criteria'=>$criteria,
));
}
public function trash($userid)
{
$this->getDbCriteria()->mergeWith(array(
'select'=>'m.message_id, m.conversation_id as
conversation, COUNT(m.message_id) AS mails_count, m.sender_id,
m.recipient_id,
m.text, m.created AS last_message_ts, ms.is_replied,
c.conversation_id, c.initiator_id,
c.interlocutor_id, c.subject, c.bm_read,
c.bm_deleted, c.modified, c.is_system',
'alias'=>'c',
'join'=>"INNER JOIN (SELECT message_id,conversation_id,
sender_id,recipient_id,text,created FROM
".MailboxModule::TBL_MSG."
WHERE sender_id=:userid OR
recipient_id=:userid ORDER BY created DESC) AS m
ON(m.conversation_id=c.conversation_id)
INNER JOIN (
SELECT conversation_id,
IF(sender_id=:userid,1,0) AS is_replied FROM
".MailboxModule::TBL_MSG."
WHERE (recipient_id=:userid OR
sender_id=:userid) ORDER BY created DESC
) AS ms ON(ms.conversation_id=c.conversation_id)",
'condition'=>'(initiator_id=:userid OR
interlocutor_id=:userid) AND
IF(initiator_id=:userid,initiator_del,interlocutor_del) > 0',
'group'=>"c.conversation_id",
'order'=>"MAX( m.created ) DESC",
'params'=>array(':userid'=>$userid,
':bminit'=>self::getBitMaskValues(self::INITIATOR_FLAG),
':bminter'=>self::getBitMaskValues(self::INTERLOCUTOR_FLAG)),
));
return $this;
}
public function inbox($userid)
{
$this->getDbCriteria()->mergeWith(array(
'select'=>'
m.message_id, m.conversation_id as conversation,
COUNT(m.message_id) AS mails_count, m.sender_id,
m.recipient_id,
m.text, m.created AS last_message_ts, ms.is_replied,
c.conversation_id, c.initiator_id,
c.interlocutor_id, c.subject, c.bm_read,
c.bm_deleted, c.modified, c.is_system',
'alias'=>'c',
'join'=>"INNER JOIN (
SELECT message_id,conversation_id,
sender_id,recipient_id,text,created FROM

```

```

".MailboxModule::TBL_MSG."
        WHERE recipient_id=:userid ORDER BY created
DESC
        ) AS m ON(m.conversation_id=c.conversation_id)
        INNER JOIN (
                SELECT conversation_id,
IF(sender_id=:userid,1,0) AS is_replied FROM
".MailboxModule::TBL_MSG."
WHERE (recipient_id=:userid OR
sender_id=:userid) ORDER BY created DESC
        ) AS ms
ON(ms.conversation_id=c.conversation_id)",
        'condition'=>'(c.initiator_id=:userid OR
c.interlocutor_id=:userid) AND (c.bm_deleted
IF(c.initiator_id=:userid,:bminit,:bminter)) = 0',
        'group'=>"c.conversation_id",
        'order'=>"MAX( m.created ) DESC",
        'params'=>array(':userid'=>$userid,
                        ':bminit'=>self::INITIATOR_FLAG,
                        ':bminter'=>self::INTERLOCUTOR_FLAG),
        ));
        return $this;
    }

    public function conversations($userid,$type='deleted')
    {
        switch ($type) {
            case 'deleted':
                $where = "c.initiator_del > 0 OR
c.interlocator_del > 0";
                break;
            default: $where = 'c.bm_deleted != 3';
        }
        $query = "SELECT m.message_id, m.conversation_id,
COUNT(m.message_id) AS mails_count, m.sender_id, m.recipient_id,
m.text, m.created AS last_message_ts, ms.is_replied,
c.conversation_id, c.initiator_id,
c.interlocutor_id, c.subject, c.bm_read,
c.bm_deleted, c.modified, c.is_system
FROM ".MailboxModule::TBL_CONV." AS c
WHERE $where";
        $sql = Yii::app()->db->createCommand($query); $sql-
>bindValue(':userid',$userid,PDO::PARAM_STR);
        $sql-
>bindValue(':bminit',self::getBitMaskValues(self::INITIATOR_FLAG),
PDO::PARAM_STR);
        $sql-
>bindValue(':bminter',self::getBitMaskValues(self::INTERLOCUTOR_FL
AG),PDO::PARAM_STR);
        //$sql->bindParam(':page',$page,PDO::PARAM_INT);
        //$sql->bindParam(':perpage',$perpage,PDO::PARAM_INT);
        $convs = $sql->queryAll();

        // or using: $rawData=User::model()->findAll();
        $dataProvider=new CArrayDataProvider($convs, array(
            'keyField'=>'conversation_id',

```

```

        'id'=>'inboxDataProvider',
        'sort'=>array(
            'attributes'=>array(
                'subject', 'last_message_ts',
'sender_id',
                // 'mails_count','initiator_id',
'interlocutor_id', 'recipient_id', 'text',
                // 'bm_read','bm_deleted', 'is_system',
'is_readable', 'is_replied', 'conversation_id',
            ),
        ),
        'pagination'=>array(
            'pageSize'=>Yii::app()->controller-
>module->pageSize,
        ),
    ));
    return $dataProvider;
}
public static function newMsgs($userid)
{
    // count messages
    $query = "SELECT SQL_CACHE COUNT(c.conversation_id) AS
num_messages FROM
        ".MailboxModule::TBL_CONV." AS c
        INNER JOIN ( SELECT message_id,conversation_id
        FROM ".MailboxModule::TBL_MSG."
        WHERE recipient_id=:userid
        ORDER BY created DESC
        ) AS m
ON(m.conversation_id=c.conversation_id)
        WHERE (c.initiator_id=:userid OR
c.interlocutor_id=:userid)
        AND (c.bm_read & IF(c.initiator_id=:userid,
:bminit, :bminter)) = 0
        AND (c.bm_deleted & IF(c.initiator_id=:userid,
:bminit, :bminter)) = 0
        GROUP BY c.conversation_id
";
    $sql = Yii::app()->db->createCommand($query);
    $sql->bindValue(':userid',$userid,PDO::PARAM_STR);
    $sql-
>bindValue(':bminit',self::INITIATOR_FLAG,PDO::PARAM_STR);
    $sql-
>bindValue(':bminter',self::INTERLOCUTOR_FLAG,PDO::PARAM_STR);
    $convs = $sql->queryAll();
    $count=0;
    foreach($convs as &$conv) {
    if ($conv['num_messages'] >= 1)
        $count++;
    }
    return $count;
}
public function deleted()
{
    $this->getDbCriteria()->mergeWith(array(
'select'=>'conversation_id, initiator_id,

```

```

interlocutor_id, c.bm_deleted, c.modified, c.is_system',
        'condition'=>'c.initiator_del > 0 OR
c.interlocutor_del > 0',
        ));
        return $this;
    }
    public static function conversation($id,$order='DESC')
    {
        if(!is_int((int)$id))                throw        new
Exception('Bad conversation Id');          $with        =
array('messages'=>array('order'=>'created
'.$order));
        return Mailbox::model()->with($with)->findByPk($id);
    }
    public function markRead($userid)
    {
        if($this->initiator_id == $userid)
            $this->bm_read = $this->bm_read |
Mailbox::INITIATOR_FLAG;
        else
            $this->bm_read = $this->bm_read |
Mailbox::INTERLOCUTOR_FLAG;
        $this->updateByPk($this-
>conversation_id,array('bm_read'=>$this->bm_read));
    }
    public function belongsTo($userid)
    {
        return ($this->initiator_id == $userid || $this-
>interlocutor_id == $userid);
    }
    public function isNew($userid)
    {
        if($this->initiator_id == $userid )
            $flag = self::INITIATOR_FLAG;
        else
            $flag = self::INTERLOCUTOR_FLAG;

        return !($this->bm_read & $flag);
    }
    /*
    * Button Actions
    */
    public function read($userid)
    {
        if($this->initiator_id == $userid) {
            $this->bm_read = $this->bm_read |
self::INITIATOR_FLAG;
        }
        elseif($this->interlocutor_id == $userid) {
            $this->bm_read = $this->bm_read |
self::INTERLOCUTOR_FLAG;
        }
        else
            throw new Exception('User denied');
        return true;
    }
}

```

```

        public function unread($userid)
        {
            if($this->initiator_id == $userid) {
                $this->bm_read = $this->bm_read &
                ~self::INITIATOR_FLAG;
            }
            elseif($this->interlocutor_id == $userid) {
                $this->bm_read = $this->bm_read &
                ~self::INTERLOCUTOR_FLAG;
            }
            else
                throw new Exception('User denied');
            return true;
        }
        public function restore($userid)
        {
            if($this->initiator_id == $userid) {
                $this->bm_deleted = $this->bm_deleted &
                ~self::INITIATOR_FLAG;
                $this->initiator_del = 0;
            }
            elseif($this->interlocutor_id == $userid) {
                $this->bm_deleted = $this->bm_deleted &
                ~self::INTERLOCUTOR_FLAG;
                $this->interlocutor_del = 0;
            }
            else throw new Exception('User
            denied'); return true;
        }
        /**
        * Delete the conversation in respect to user's perspective.
        When both initiator and interlocutor have
        * permanently deleted the conversation, only then is the
        conversation actually removed from the
        * database. If 'trashbox' is enabled in the module configs then
        instead of marking the conversation
        * as permanently deleted by that user, the conversation is moved
        to the trash folder by setting
        * {perspective}_del field to the day of the half-year "date('z')
        / 2" that the message was deleted.
        * Also if 'trashbox' is enabled then the cron method in
        MailboxModule must be ran at least once
        * daily in order to check the number of days the message
        has been in the trash folder, and delete it *
        permanently if days is more than config var
        'recyclePeriod'.
        *
        * @param integer $userid the user's Id in which to delete the
        conversation in respect to. The parameter
        * declaration is given a default value in order to overload
        the parents delete method and ensure that
        * conversations are deleted via the destroy method.
        * @return boolean true on success
        */
        public function delete($userid=0)
        {

```

```

        //die((int)Yii::app()->controller->module-
>recyclePeriod);
        if(!$userid)
            throw new Exception('User Id must be supplied for delete
method');
        $day = date("z") + 1;
if($day > 183)                $day
= $day - 183;
        if($this->initiator_id == $userid) {
            // mark deleted
            $this->bm_deleted = $this->bm_deleted |
self::INITIATOR_FLAG;
            // check if in trash already, or if trashbox is
disabled
                if((!$this->initiator_del && $this-
>initiator_del > Yii::app()->controller->module->recyclePeriod)
                || !Yii::app()->controller->module-
>trashbox) {
                    // permanently delete
                    $this->initiator_del = null;
                } else {
                    // move to trash
                    $this->initiator_del = $day;
                }
            }
        elseif($this-
>interlocutor_
id == $userid)
        {
            // mark deleted
            $this->bm_deleted = $this->bm_deleted |
self::INTERLOCUTOR_FLAG;
            // check if in trash already, or if trashbox is
disabled
                if($this->interlocutor_del > Yii::app()-
>controller->module->recyclePeriod || !Yii::app()->controller-
>module->trashbox) {
                    // permanently delete
                    $this->interlocutor_del = null;
                } else {
                    // move to trash
                    $this->interlocutor_del = $day;
                }
            }
        } else
            throw new Exception('User denied');

        // if both parties have permanently deleted conversation
        if(is_null($this->initiator_del)                &&
is_null($this->interlocutor_del))
        {
            // delete conversation and messages from database
            $this->destroy();
        }
        return true;
    }
}

```

```

/*
 * End button actions
 */
/**
 * This method is used by the cron method in the
 * MailboxModule class to determine when a conversation
 * should be marked as permanently deleted by a user
 * depending on the recycling period defined in the
 * module configs by the attribute 'recyclePeriod'. And if both
 * users have the conversation marked
 * as permanently deleted then remove the conversation from the
 * database.
 *
 * @param integer $mask
 * @param boolean $string_presentation
 * @return string|array
 */
public function recycle()
{
    // check initiator trash
    if($this->initiator_del > 0)
    {
        // if not within recycling period
        if(!self::withinDays(Yii::app()->controller-
>module->recyclePeriod,
                                $this->initiator_del)) {
            // mark permanently deleted
            $this->initiator_del = null;
        }
    }
    // check interlocutor trash
    if($this->interlocutor_del > 0)
    {
        // if not within recycling period
        if(!self::withinDays(Yii::app()->controller-
>module->recyclePeriod,
                                $this->interlocutor_del)) {
            // mark permanently deleted
            $this->interlocutor_del = null;
        }
    }
    // if both marked as permanently deleted
    if(is_null($this->initiator_del) &&
is_null($this->interlocutor_del))
    {
        // delete this conversation from database
        $this->destroy();
    }
    else{
        // otherwise save possible changes
        $this->save();
    }
}
public function destroy()
{
    // delete all conversation messages

```

```

        $msg_count=Message::model()-
>deleteAll('conversation_id = :cid',array(':cid'=>$this-
>conversation_id));
        if(!$msg_count)
            throw new Exception('Conversation messages not
found?');
        // delete conversation
        return parent::delete();
    }
/**
 * Get array of values that make up mask.
 *
 * @param integer $mask
 * @param boolean $string_presentation
 * @return string|array
 */
    public static function getBitMaskValues( $mask,
$string_presentation = true )
    {
        $available_mask_arr = array(self::INITIATOR_FLAG,
self::INTERLOCUTOR_FLAG);
        if ( in_array( $mask, $available_mask_arr ) )
        {
            $values_count = pow( 2, 2 );
            if ( $string_presentation )
            {
                $return_arr = "";
                for ( $i = 1; $i < $values_count; $i++ )
                    if ( ($mask & $i) == $mask )
                        $return_arr .= $i.", ";
                return substr( $return_arr, 0, -2 );
            }
            else
            {
                $return_arr = array();
                for ( $i = 1; $i < $values_count; $i++ )
                    if ( ($mask & $i) == $mask )
                        $return_arr[$i] = $i;
                return $return_arr;
            }
        }
    }
    public static function withinDays($period,$deleted_on)
    {
        $day = date("z") + 1;
        if($day > 183)
            $day
= $day - 183;
        elseif($day < $deleted_on)
            $day += 183;
        if( ($day - $deleted_on) <= $period )
            return true;
        return false;
    }
}

```

## Message.php

```
<?php
/**
 * This is the model class for table "{{mailbox_message}}". *
 * The followings are the available columns in table
 * "{{mailbox_message}}":
 * @property string $message_id
 * @property string $conversation_id
 * @property string $created
 * @property string $sender_id
 * @property string $recipient_id
 * @property string $text
 * @property string $is_kiss
 * @property string $is_readable
 * @property string $hash
 */
class Message extends
 CActiveRecord
{
    public          $modified;
    public          $subject;
    public          $is_replied;
    public $initiator_id;
    public $bm_read;
}
/**
 * Returns the static model of the specified AR class.
 * @param string $className active record class name.
 * @return Message the static model class
 */
public static function model($className=__CLASS__)
{
    return parent::model($className);
}
/**
 * @return string the associated database table name
 */
public function tableName()
{
    return MailboxModule::TBL_MSG;
}
/**
 * @return array validation rules for model attributes.
 */
public function rules()
{
    $module =& Yii::app()->controller->module;
    $p = new CHtmlPurifier();
    if(Yii::app()->controller->module->isAdmin())
        $p->options = $module->adminHtml;
    else
        $p->options = $module->html;
    return array(
        array('text', 'required'),
        array('created, sender_id, recipient_id',
            'length', 'max'=>10),
        array('text','filter','filter'=>array($p,'purify')),
    );
}
```

```

        array('file_foto', 'file', 'types'=>'doc, docx,
pdf, rar, zip, 7z', 'allowEmpty' => true),
        // The following rule is used by search().      // Please
remove those attributes that should not be searched.
        array('message_id, conversation_id, created,
sender_id, recipient_id, text, hash, file_foto', 'safe',
'on'=>'search'),
    );
}
/**
 * @return array relational rules.
 */
public function relations()
{
    // NOTE: you may need to adjust the relation name and the
related
    // class name for the relations automatically generated
below.
    return array(
        'conversation'=>array(self::HAS_ONE,
'Mailbox','conversation_id'),
    );
}
/**
 * @return array customized attribute labels (name=>label)
 */
public function attributeLabels()
{
    return array(
        'message_id' => 'Message',
        'conversation_id' => 'Conversation Id',
        'created' => 'Time Stamp',
        'sender_id' => 'Sender',
        'recipient_id' => 'Recipient',
        'text' => 'Text',
        'file_foto' => 'File',
    );
}
/**
 * Retrieves a list of models based on the current
search/filter conditions.
 * @return CActiveDataProvider the data provider that can
return the models based on the search/filter conditions.
 */
public function search()
{
    // Warning: Please modify the following code to remove
attributes that
    // should not be searched.
    $criteria=new CDbCriteria;
    $criteria->compare('message_id',$this-
>message_id,true);
    $criteria-
>compare('conversation_id',$this->conversation_id,true);
    $criteria->compare('created',$this->created,true);
    $criteria->compare('sender_id',$this->sender_id,true);

```

```

        $criteria->compare('recipient_id',$this-
>recipient_id,true);
        $criteria->compare('text',$this->text,true);
        $criteria->compare('file_foto',$this->file_foto,true);
        return new CActiveDataProvider($this, array(
            'criteria'=>$criteria,
        ));
    }
    public function conversation($conversation_id)
    {
        if(!preg_match('/^[0-9]+$/', $conversation_id))
            die('Conversation Id must be an
integer:'.$conversation_id);
        $this->getDbCriteria()->mergeWith(array(
            'select'=>'*',
            'condition'=>'conversation_id = :cid',
            'params'=>array(':cid'=>$conversation_id)
        ));
        return $this;
    }
    public function sent($userid)
    {
        $this->getDbCriteria()->mergeWith(array(
            'select'=>'m.message_id, m.sender_id,
m.recipient_id,
                m.text, m.created,
                c.conversation_id, c.initiator_id ,
c.interlocutor_id, c.subject, c.bm_read,
                c.bm_deleted, c.modified, c.is_system,
                ms.created as is_replied',
            'alias'=>'m',
            'join'=>"
                INNER JOIN ".MailboxModule::TBL_CONV." AS c
ON(c.conversation_id=m.conversation_id)
                LEFT JOIN (
                    SELECT conversation_id, created FROM
                ".MailboxModule::TBL_MSG."
                    WHERE recipient_id=:userid AND created
                ) AS ms ON(ms.conversation_id=m.conversation_id
AND ms.created > m.created)
                ",
            'condition'=>'sender_id=:userid',
            'order'=>"m.created DESC",
            'params'=>array(':userid'=>$userid),
        ));
        return $this;
        /* old */
        $is_replied = "
            INNER JOIN (
                SELECT conversation_id,
IF(sender_id=:userid,'yes','no') AS is_replied FROM
                ".MailboxModule::TBL_MSG."
                WHERE (recipient_id=:userid OR
sender_id=:userid) ORDER BY created DESC
            ) AS ms

```

```

ON(ms.conversation_id=c.conversation_id)";
    }
    public static function crc64($text)
    {
        $crc64 = sprintf('%u',hash('crc32', $text))
        . sprintf('%u',hash('crc32b', $text));
        return base_convert($crc64,16,10); // 64bit INT
    }
    public function isRead($userid)
    {
        if($this->is_replied)
            return 1;
        if($this->initiator_id == Yii::app()->controller>module->
        >getUserId() )
            $flag = Mailbox::INTERLOCUTOR_FLAG;
        else
            $flag = Mailbox::INITIATOR_FLAG;
        return $this->bm_read & $flag;
    }
}

```

#### **News.php**

```

<?php
/**
 * This is the model class for table "{{mailbox_conversation}}".
 *
 * The followings are the available columns in table
 * '{{mailbox_conversation}}':
 * @property string $conversation_id
 * @property integer $initiator_id
 * @property integer $interlocutor_id
 * @property string $subject
 * @property integer $bm_read
 * @property integer $bm_deleted
 * @property string $modified
 * @property string $is_system
 */
class News extends Mailbox
{
    public static function model($className=__CLASS__)
    {
        return parent::model($className);
    }
    public function isNew($userid)
    {
        /*
 * If your database keeps track of user logins date/time you could
rewrite
 * this method to return true if the modified timestamp is greater
than
 * (ie. more recent) the user's previous login
timestamp (ie the last time
 * the user successfully logged-in, not counting the
current login).
 */
        /*
if($this->modified > Yii::app()->user->previousLogin )

```

```

        return true;
    return false;
    */

    // for now we just use a static time limit...
    if($this->modified > time() - 3600 * 24 * 7)
        return true;
    return false;
}
public function delete()
{
    return $this->destroy();
}
}

```

### **b.View mailbox.php**

```

<?php
if($this->getAction()->getId()!='inbox')
$this->breadcrumbs=array(
    ucfirst($this->module->id)=>array('inbox'),
    ucfirst($this->getAction()->getId())
); else
    $this->breadcrumbs=array($this->module->id);

$this->renderpartial('_menu');
if(isset($_GET['Message_sort']))
    $orderby = $_GET['Message_sort'];
elseif(isset($_GET['Mailbox_sort']))
    $orderby = $_GET['Mailbox_sort'];
else
    $orderby = '';
$ie6br = <<<EOD
<!--[if lt IE 6]>
<br clear="all" />
<![endif]--> EOD;
echo '<div id="mailbox-list" class="mailbox-list ui-
helperclearfix" sortby="'. $orderby.' ">';

$this->renderpartial('_flash');
if($dataProvider->getItemCount() > 0)
{ ?>
<form id="message-list-form" action="<?php echo
$this->createUrl($this->getId().'/'.$this->getAction()->getId());
?>" method="post">
<input type="hidden" class="mailbox-count" name="ui[]"
value="<?php echo $dataProvider->getItemCount(); ?>" />
<input type="hidden" class="mailbox-sortby" name="ui[]"
value="<?php echo $orderby; ?>" />
<div class="mailbox-clistview-container ui-helper-clearfix">
<?php
if($dataProvider->getItemCount() > 1 && $this->getAction()-
>getId() != 'sent') : ?>
<div class="btn-group mailbox-checkall-buttons">
<button class="checkall" />Check All</button>
<button class="uncheckall" />Uncheck
All</button>

```

```

        </div>
<?php
endif;
$this->widget('zii.widgets.CListView', array(
    'id'=>'mailbox',
    'dataProvider'=>$dataProvider,
    'itemView'=>'_list',
    'itemsTagName'=>'table',
    'template'=>'<div class="mailbox-
summary">{summary}</div>{sorter}'. $ie6br.'<div id="mailbox-items"
class="ui-helper-clearfix">{items}</div>{pager}',
    'sortableAttributes'=>$this->getAction()->getId()=='sent'?
array('created'=>'Date Sent') : array('modified'=>'Date
Received'),
    'loadingCssClass'=>'mailbox-loading',
    'ajaxUpdate'=>'mailbox-list',
    'afterAjaxUpdate'=>'$.yiimailbox.updateMailbox',
    'emptyText'=>'<div style="width:100%"><h3>You have no mail in
your '. $this->getAction()->getId().' folder.</h3></div>',
    //'htmlOptions'=>array('class'=>'ui-helper-clearfix'),
    'sorterHeader'=>'',
    'sorterCssClass'=>'mailbox-sorter',
    'itemsCssClass'=>'mailbox-items-tbl ui-helper-clearfix',
    'pagerCssClass'=>'mailbox-pager',
    //'updateSelector'=>'.inbox',
));
?>
<?php if($this->getAction()->getId()!='sent') : ?><div
style="clear:left"><span class="mailbox-buttons-label">With
selected:</span>
    <?php if($this->getAction()->getId()=='trash') : ?>
<input type="submit" id="mailbox-action-restore" class="btn
mailbox-button" name="button[restore]" value="restore" />
<input type="submit" id="mailbox-action-delete" class="btn
mailbox-button" name="button[delete]" value="delete forever" />
    <?php else: ?>
        <?php if(!$this->module->readOnly || ( $this-
>module->readOnly && !$this->module->isAdmin())) : ?> <input
type="submit" id="mailbox-action-delete" class="btn mailbox-
button" name="button[delete]" value="delete" />
        <?php endif; ?>
        <input type="submit" id="mailbox-action-read" class="btn
mailbox-button" name="button[read]" value="read" /> <input
type="submit" id="mailbox-action-unread" class="btn
mailbox-button" name="button[unread]" value="unread" />
        <?php endif; ?>
</div>
    <?php endif; ?>
</div>
</form>
<?php
}
else {
    $this->renderpartial('_empty');
}
?>
</div>

```

```

<script type="text/javascript">
/** / jQuery(function($
{
    $(' .message-subject').hide();
});
/*]]&gt;*/
&lt;/script&gt;
<b>__mailbox.php</b>
&lt;?php
if(isset($_GET['Message_sort']))
    $sortby = $_GET['Message_sort'];
elseif(isset($_GET['Mailbox_sort']))
    $sortby = $_GET['Mailbox_sort'];
else
    $sortby = '';
echo '&lt;div id="mailbox-list" class="mailbox-list ui-
helperclearfix" sortby="'. $sortby.' "&gt;';

$this-&gt;renderpartial('_flash');
$ie6br = &lt;&lt;&lt;EOD
&lt;!--[if lt IE 6]&gt;
&lt;br clear="all" /&gt;
&lt;![endif]--&gt;
EOD; if($dataProvider-&gt;getItemCount()
&gt; 0) {
?&gt;
&lt;form id="message-list-form" action="&lt;?php echo
$this-&gt;createUrl($this-&gt;getId().'/'.$this-&gt;getAction()-&gt;getId());
?&gt;" method="post"&gt;
    &lt;input type="hidden" class="mailbox-count" name="ui[]"
value="&lt;?php echo $dataProvider-&gt;getItemCount(); ?&gt;" /&gt;
    &lt;input type="hidden" class="mailbox-sortby" name="ui[]"
value="&lt;?php echo $sortby; ?&gt;" /&gt;
    &lt;div class="mailbox-clistview-container"&gt;
    &lt;?php
        if($dataProvider-&gt;getItemCount() &gt; 1 &amp;&amp; $this-&gt;getAction()-
&gt;getId() != 'sent') : ?&gt;
            &lt;div class="btn-group mailbox-checkall-buttons"&gt;
                &lt;button class="btn checkall"&gt;Check All&lt;/button&gt;
                &lt;button class="btn uncheckall"&gt;Uncheck
All&lt;/button&gt;
            &lt;/div&gt;
        &lt;?php
    endif;

$this-&gt;widget('zii.widgets.CListView', array(
    'id'=&gt;'mailbox',
    'dataProvider'=&gt;$dataProvider,
    'itemView'=&gt;'_list',
    'itemsTagName'=&gt;'table',
    'template'=&gt;'&lt;div class="mailbox-
summary"&gt;{summary}&lt;/div&gt;{sorter}'. $ie6br.'&lt;div id="mailbox-items"
class="ui-helper-clearfix"&gt;{items}&lt;/div&gt;{pager}',
</pre>
</div>
```

```

        'sortableAttributes'=>$this->getAction()->getId()=='sent'?
array('created'=>'Date Sent') : array('modified'=>'Date
Received'), 'loadingCssClass'=>'mailbox-loading',
    'ajaxUpdate'=>'mailbox-list',
    'afterAjaxUpdate'=>'$.yiimailbox.updateMailbox',
    'emptyText'=>'<div style="width:100%"><h3>You have no mail in
your '.$this->getAction()->getId().' folder.</h3></div>',
    //'htmlOptions'=>array('class'=>'ui-helper-clearfix'),
    'sorterHeader'=>'',
    'sorterCssClass'=>'mailbox-sorter',
    'itemsCssClass'=>'mailbox-items-tbl ui-helper-clearfix',
    'pagerCssClass'=>'mailbox-pager',
    //'updateSelector'=>'.inbox',
    );
?>
<?php if($this->getAction()->getId()!='sent') : ?><div
style="clear:left"><span class="mailbox-buttons-label">With
selected:</span>
    <?php if($this->getAction()->getId()=='trash') : ?>
<input type="submit" id="mailbox-action-restore" class="btn
mailbox-button" name="button[restore]" value="restore" />
<input type="submit" class="btn" name="button[delete]"
value="delete forever" />
    <?php else: ?>
        <?php if(!$this->module->readOnly || ( $this-
>module->readOnly && !$this->module->isAdmin() ): ?> <input
type="submit" id="mailbox-action-delete" class="btn
mailbox-button" name="button[delete]" value="delete" />
        <?php endif; ?>
        <input type="submit" id="mailbox-action-read" class="btn mailbox-
button" name="button[read]" value="read" />
        <input type="submit" id="mailbox-action-unread" class="btn
mailbox-button" name="button[unread]" value="unread" />
        <?php endif; ?>
</div>
    <?php endif; ?>
</div>
</form>
<?php
} else
{
}
?>
</div>

<script type="text/javascript">
/**/ jQuery(function($
{
    $(' .message-subject').hide();
});
/*]]&gt;*/
&lt;/script&gt;
<b><u>_list.php</u></b>
&lt;?php

$userid = $this-&gt;module-&gt;getUserId();
</pre>
</div>
```

```

    if($this->getAction()->getId()=='sent')
    {
        $counterUserId = $data->recipient_id;
    } else {
        if($this->module->getUserId() == $data->
        >initiator_id)
            $counterUserId = $data->interlocutor_id;
        else
            $counterUserId = $data->initiator_id;
    }
    $username = $this->module->getFromLabel($counterUserId);

    if($username && ($this->module->isAdmin() || $this->module-
    >linkUser)) {
        $url = $this->module->getUrl($counterUserId);
        if($url)
            $username = '<a href="'. $url. '">'. $username. '</a>';
    }
    elseif(!$username)
        $username = '<span class="mailbox-deleted-user">'. $this->module-
        >deletedUser. '</span>';

    $viewLink = $this->createUrl('message/view', array('id'=>$data-
    >conversation_id));

    if($this->getAction()->getId()=='sent') {
        $received = $this->module->getDate($data->created);
        if($this->module->recipientRead)
            $itemCssClass = ($data->isRead($userid))? 'msg-read' :
            'msg-deliver';
        else
            $itemCssClass = 'msg-sent';
    } else{
        $received = $this->module->getDate($data->modified);
        $itemCssClass = $data->isNew($userid)? 'msg-new' : 'msgread';
    }
    switch($itemCssClass)
    {
        case 'msg-read': $status = ($this->getAction()-
        >getId()=='sent')? 'Recipient has read your message' : 'Message
        has been read' ; break;
        case 'msg-deliver': $status =
        'Recipient has not read your message yet';
        case 'msg-new':
        $status = ($this->getAction()-
        >getId()=='sent')? 'Recipient has not read your message yet' :
        'You received a new message'; break;
        case 'msg-sent': $status = "You sent message {$username} a
        message";
    }
    $subject = '<span class="mailbox-subject-text">';
    $subject .= '<a class="mailbox-link" title="'. $status. '"
    href="'. $viewLink. '">';
    $subjectSeperator = ' - ';
    if(strlen($data->subject) > $this->module->subjectMaxCharsDisplay)
    {
        $subject .= substr($data->subject,0,$this->module-
        >subjectMaxCharsDisplay - strlen($this->module->ellipsis) ).
        $this->module->ellipsis . '</a></span>';
    } else

```

```

{
    $subject .= $data->subject . '</a></span><span
class="mailbox-msg-brief">'. $subjectSeperator
    . substr(strip_tags($data->text), 0, $this-
>module->subjectMaxCharsDisplay - strlen($data->subject) -
strlen($subjectSeperator) - strlen($this->module->ellipsis) );
if(strlen($data->subject) + strlen($data->text) +
strlen($subjectSeperator) > $this->module->subjectMaxCharsDisplay)
    $subject .= $this->module->ellipsis;
}
$subject = preg_replace('/[\n\r]+/', '', $subject);
$subject.= '</span>';
?>
<tr class="mailbox-item <?php echo $itemCssClass; ?><?php
if($this->getAction()->getId()!='sent') echo 'mailbox-
draggablerow'; ?>">
<?php if($this->getAction()->getId()!='sent'): // add dragdrop
handle ?>
        <td width="25" style="width:50px;"><div
class="mailbox-item-wrapper mailbox-drag">&nbsp;</div></td>
        <?php endif; ?>

<td width="25" style="width:50px;">
        <?php if($this->getAction()->getId()=='sent') : ?>
            <div class="mailbox-item-wrapper">&nbsp;</div>
        <?php else: ?>
            <div class="mailbox-item-wrapper">
                <label class="ui-helper-reset" for="conv_<?php echo
$data->conversation_id; ?>">
                <div class="mailbox-check mailbox-ellipsis">
                    <input class="mailbox-check " id="conv_<?php
echo $data->conversation_id; ?>" type="checkbox" name="convs[]"
value="<?php echo $data->conversation_id; ?>" />
                </div>
            </div>
            </label>
        <?php endif; ?>
</td>
<td width="100" style="width:200px;">
<div class="mailbox-item-wrapper mailbox-from mailbox-
ellipsis"><?php echo $username; ?></div>
</td>
<td width="100%" class="mailbox-subject-brief"> <div
class="mailbox-item-wrapper mailbox-item-outer mailbox-
subject"><div class="mailbox-item-inner mailbox-ellipsis ui-
helper-clearfix">
            <?php echo $subject; ?>
        </div></div>
</td>
<td width="150" class="mailbox-received">
        <div align="right" class="mailbox-item-wrapper"
style="width:80px">
            <?php if($data->is_replied) : ?>
                <div class="mailbox-replied" title="this message
has been replied to">&nbsp;&nbsp;</div>
            <?php endif; ?>

```

```

                <?php echo $received; ?>
            </div>

</td>
</tr>_men
u.php
<?php
$newMsgs = $this->module->getNewMsgs(); $action
= $this->getAction()->getId();
    if($this->module->authManager)
    {
        $authNew = Yii::app()->user>checkAccess("Mailbox.Message.New");
        $authInbox
            =
                Yii::app()-
            >user>checkAccess("Mailbox.Message.Inbox");
        $authSent = Yii::app()->user>checkAccess("Mailbox.Message.Sent");
        $authTrash
            =
                Yii::app()-
            >user>checkAccess("Mailbox.Message.Trash
"); }
    else
    {
        $authNew = $this->module->sendMsgs && (!$this->module-
>readOnly || $this->module->isAdmin());
        $authInbox = ( !$this->module->readOnly || $this->module-
>isAdmin() );
        $authTrash = $this->module->trashbox && (!$this->module-
>readOnly || $this->module->isAdmin());
        $authSent = $this->module->sentbox && (!$this->module>readOnly ||
$this->module->isAdmin());
    }
?>
<div class="mailbox-menu ui-helper-clearfix">
    <div class="mailbox-menu-folders ui-helper-clearfix">
        <?php
            if($authInbox):?>
                <div id="mailbox-inbox" class="mailbox-menu-item <?php
echo ($action=='inbox')? 'mailbox-menu-current' : '' ; ?>">
                    <a href="<?php echo $this-
>createUrl('message/inbox'); ?>">Inbox <span class="mailbox-
newmsgs"><?php echo $newMsgs? '($newMsgs.)' : null ;
?></span></a>
                </div>
            <?php endif;
            if($authSent) : ?>
                <div id="mailbox-sent" class="mailbox-menu-item <?php
if($action=='sent') echo 'mailbox-menu-current ' ; ?>">
                    <a href="<?php echo $this-
>createUrl('message/sent'); ?>">Sent Mail</a>
                </div>
            <?php endif;
            if($authTrash) : ?>
                <div id="mailbox-trash" class="mailbox-menu-item <?php
if($action=='trash') echo 'mailbox-menu-current ' ; ?>">
                    <a href="<?php echo $this-
>createUrl('message/trash'); ?>">Trash </a>
                </div>
            <?php endif; ?>

```

```

        </div>
<?php if($authNew)
:
    ?>
<div class="mailbox-menu-newmsg ui-helper-clearfix"
align="center">
    <span><a href="<?php echo $this-
>createUrl('message/new'); ?>">New Message</a></span>
</div>
<?php endif; ?>
</div>

```

### **compose.php**

```

<?php
$this->breadcrumbs=array(
ucfirst($this->module->id)=>array('inbox'),
ucfirst($this->getAction()->getId()) );
$this->renderpartial('_menu');
?>
<div class="mailbox-compose ui-helper-clearfix">

<?php

$this->renderPartial('_flash');

$form=$this->beginWidget('CActiveForm', array(
'id'=>'message-form',
'enableAjaxValidation'=>false,
'htmlOptions'=>array('autocomplete'=>$this-
>createUrl('ajax/auto')),
'htmlOptions'=>array('enctype'=>'multipart/form-data'),
)); ?>
    <div class="form message-create">

        <div class="mailbox-input-row">
            <div style="float:left">
                <?php echo
CHtml::activeLabelEx($conv,'to'); ?>
            </div>
            <div style="margin-left:80px">
                <?php echo $form-
>textField($conv,'to',array('style'=>'width:100%;','id'=>'messaget
o','class'=>'mailbox-input', 'edit'=>$this->module->editToField?
'1' : null)); ?>
                <?php echo $form->error($conv,'to'); ?>

            <?php

                if($this->module->userSupportList)
                {

                    $reps = $this->module-
>getUserSupportList();
                    echo '<select name="ajax[to]"
class="mailbox-support-list" edit="'.(($this->module-

```

```

>editToField)? '1' : null).'" >';
foreach($reps as $key =>
&$label)
    {
    ?>
    <option type="hidden"
value="<?php echo $key; ?>"><?php echo $label; ?></option>
    <?php
    }
    echo '</select>';
    }
    ?>
    </div>
</div>
<div class="mailbox-input-row">
    <div style="float:left">
        <?php echo
CHtml::activeLabelEx($conv, 'subject', array('class'=>'mailboxlabel'
)); ?>
        </div>
        <div class="mailbox-compose-inputwrap"
style="margin-left:80px;">
            <?php echo $form-
>textField($conv, 'subject', array('class'=>'mailboxinput', 'style'=>
'width:100%;', 'placeholder'=>$this->module-
>defaultSubject)); ?>
            <?php echo $form->error($conv, 'subject');
?>
        </div>
        <div style="float:left">
            <?php echo
CHtml::activeLabelEx($msg, 'file_foto', array('class'=>'mailboxlabel
')); ?>
            </div>
            <div class="mailbox-compose-inputwrap"
style="margin-left:80px;">
                <?php echo $form-
>fileField($msg, 'file_foto', array('size'=>30, 'maxlength'=>30)); ?>
                <?php echo $form->error($msg, 'file_foto');
?>
            </div>
        </div>
        <div class="mailbox-textarea-wrap">
            <?php echo $form-
>textArea($msg, 'text', array('cols'=>50, 'rows'=>7,
'class'=>'mailbox-message-
input', 'style'=>'width:100%;', 'placeholder'=>'Enter      message
here...')); ?>
            <?php echo $form->error($msg, 'text'); ?>
            </div>
            <div>
                <button      class="btn      btn-large      message-btn-reply">Send
Message</button>
            </div>
        </div>
<?php $this->endWidget(); ?><!-- form -->

```

```

</div>
<!-- mailbox -->

message.php
<?php
$this->breadcrumbs=array(    ucfirst($this->module-
>id)=>array('mailbox/inbox'),
    'Message',
);

$this->renderPartial('_menu');

$subject    =    ($conv->subject)?    $conv->subject    :    $this-
>module>defaultSubject;
if(strlen($subject) > 100)
{
    $subject = substr($subject,0,100);
}

?>
<div class="mailbox-message-list">

<?php
    foreach(Yii::app()->user->getFlashes() as $key => $message) {
echo '<div class="flash-' . $key . '">' . $message .
"</div>\n";
    }
?>
<div class="mailbox-message-subject mailbox-ellipsis"><?php echo
$subject; ?></div>

<br />
<?php
$first_message=1; foreach($conv->messages
as $msg):
    $sender = $this->module->getUserName($msg->sender_id);
    if(!$sender)
        $sender = $this->module->deletedUser;
    ?>

    <div class="mailbox-message-header">
        <div class="message-sender">
<?php echo ($msg->sender_id == Yii::app()->user->id)? 'You' :
ucfirst($sender);        echo ($first_message)? ' said' : '
replied'; ?></div>        <div class="message-date"><?php echo
date("Y-m-d H:i a",$msg->created); ?></div>
        <br />
    </div>
    <div class="mailbox-message-text"><?php echo $msg->text;
?></div>
    <br />
<?php $first_message=0; endforeach;

if($this->module->authManager)

```

```

    $authReply = Yii::app()-
    >user>checkAccess("Mailbox.Message.Reply");
    else
        $authReply = $this->module->sendMsgs;

    if($authReply)
    {

        $form=$this->beginWidget('CActiveForm', array(
            'action'=>$this-
        >createUrl('message/reply',array('id'=>$_GET['id'])),
            'id'=>'message-reply-form',
            'enableAjaxValidation'=>false,
        )); ?>
        <div class="mailbox-message-reply ui-helper-clearfix">
            <?php /* echo $form->errorSummary(array($reply,$conv));*/ ?>
            <?php echo $form->error($reply,'text'); ?>
            <div class="mailbox-textarea-wrap ui-helper-clearfix">
                <textarea name="text" cols="50" rows="7"
placeholder="Reply here..."></textarea>
            </div>
            <input type="submit" class="btn btn-large mailbox-input"
value="Send Reply" />
            </div>

        <?php $this->endWidget();
    }
    ?>
    <?php
    $this->widget('zii.widgets.grid.CGridView', array(
        'id'=>'tbl-mailbox-message-grid',
        'dataProvider'=>$msg->search(),
        'filter'=>$msg, 'columns'=>array(
            array(
                'header'=>'Attachment',
                'type'=>'raw',
                'value'=>'CHtml::link($data->file_foto,
"http://" . $_SERVER["SERVER_NAME"] . Yii::app()->request->baseUri
. "/images/foto/" . $data->file_foto)',
            ),
        ),
    )); ?>
</div>

```

### **c. Controller**

#### **MessageController.php**

```

<?php
class MessageController extends Controller
{
    public $defaultAction =
    'inbox';

    public $buttons = array( // Ex output: {count} messages have been
{value}

```

```

'default'=>array('delete'=>'deleted','read'=>'marked
read','unread'=>'marked unread'),
        'trash'=>array('delete'=>'permanently
deleted','restore'=>'restored')
    );

    public function filters()
    {
        if($this->module->authManager=='rights') {
            return array(
                'rights', // perform access control for
CRUD operations
            );
        }
        else return array();
    }

    public function behaviors()
    {
        return array(
            'ButtonAction'=>array(

                'class'=>'mailbox.behaviors.ButtonActionBehavior',
                'controller'=>$this,
                'module'=>$this->module,
                'buttons'=>$this->buttons,
                'arclass'=>'Mailbox',
            )
        );
    }

    public function actionInbox($ajax=null)
    {
        $this->module->registerConfig($this->getAction()->getId());
        $cs = $this->module->getClientScript();
        $cs->registerScriptFile($this->module-
>getAssetsUrl().'/js/mailbox.js',CClientScript::POS_END);
        //$js = '$("#mailbox-list").yiiMailboxList('.$this->module-
>getOptions().');console.log(1)';

        //$cs->registerScript('mailbox-
js',$js,CClientScript::POS_READY);

        if(isset($_POST['convs']))
        {
            $this->buttonAction('inbox');
        }

        $dataProvider = new CActiveDataProvider( Mailbox::model()-
>inbox($this->module->getUserid()) );
        if(isset($ajax))
            $this-
>renderPartial('_mailbox',array('dataProvider'=>$dataProvider));
        else{
            if(!isset($_GET['Mailbox_sort']))
                $_GET['Mailbox_sort'] = 'modified.desc';

```

```

        $this-
>render('mailbox',array('dataProvider'=>$dataProvider));
    }
}

public function actionSent()
{
    // auth manager
    if(!$this->module->authManager && (!$this->module-
>sentbox || ($this->module->readOnly && !$this->module-
>isAdmin()) ) )
        $this->redirect(array('message/inbox'));

    $this->module->registerConfig($this->getAction()->getId());

    $this->module->getClientScript()-
>registerScriptFile($this->module-
>getAssetsUrl().' /js/jquery.colors.js');
    $this->module->getClientScript()-
>registerScriptFile($this->module-
>getAssetsUrl().' /js/mailbox.js',CClientScript::POS_END);
    if(isset($_POST['convs']))
    {
        $this->buttonAction('sent');
    }
    $dataProvider = new CActiveDataProvider(
Message::model()->sent($this->module->getUserid()) );
    $this-
>render('mailbox',array('dataProvider'=>$dataProvider));
}

public function actionTrash($ajax=null)
{
    // auth manager
    if(!$this->module->authManager && (!$this->module->trashbox
|| ($this->module->readOnly && !$this->module-
>isAdmin()) ) )
        $this->redirect(array('message/inbox'));

    $this->module->registerConfig($this->getAction()->getId());

    $this->module->getClientScript()-
>registerScriptFile($this->module-
>getAssetsUrl().' /js/jquery.colors.js');
    $this->module->getClientScript()-
>registerScriptFile($this->module-
>getAssetsUrl().' /js/mailbox.js',CClientScript::POS_END);
    if(isset($_POST['convs']))
    {
        $this->buttonAction('trash','trash');
    }
    $period =& $this->module->recyclePeriod;
    Yii::app()->user->setFlash('notice', "Messages in the
trash are deleted within {$period} days.");
}

```

```

        $dataProvider = new CActiveDataProvider( Mailbox::model() -
>trash($this->module->getUserId()) );                if(isset($ajax))
            $this-
>renderPartial('_mailbox',array('dataProvider'=>$dataProvider));
            else{
                $this-
>render('mailbox',array('dataProvider'=>$dataProvider));
            }
        }

        public function actionNew()
        {
            $this->module->registerConfig($this->getAction()->getId());
            $cs = $this->module->getClientScript();
            $cs->registerScriptFile($this->module-
>getAssetsUrl().'/js/compose.js');
            $cs->registerScriptFile($this->module-
>getAssetsUrl().'/js/jquery.combobox.contacts.js');
            $js = '$(".mailbox-
compose").yiiMailboxCompose($(".$this->module->getOptions().");";
            $cs->registerScript('mailbox-
js',$js,CClientScript::POS_READY);
            if(!$this->module->authManager && (!$this->module->sendMsgs
|| ($this->module->readOnly && !$this->module-
>isAdmin())) )
                $this->redirect(array('message/inbox'));

            if(isset($_POST['Mailbox']['to']))
            {
                $t = time();
                $conv = new Mailbox();
                $conv->subject = ($_POST['Mailbox']['subject'])?
$_POST['Mailbox']['subject'] : $this->module->defaultSubject;
                $conv->to = $_POST['Mailbox']['to'];
                $conv->initiator_id = $this->module-
>getUserId();

                // Check if username exist
                if(strlen($_POST['Mailbox']['to'])>1)
                    $conv->interlocutor_id = $this->module-
>getUserId($_POST['Mailbox']['to']);
                else
                    $conv->interlocutor_id = 0;
                // ...if not check if To field is user id
                if(!$conv->interlocutor_id)
                {
                    if($_POST['Mailbox']['to'] && ($this-
>module->allowLookupById || $this->module->isAdmin()))
                        $username = $this->module-
>getUserName($_POST['Mailbox']['to']);
                    if(@$username) {
                        $conv->interlocutor_id =
$_POST['Mailbox']['to'];
                        $conv->to = $username;
                    }
                }
                else {

```

```

// possible that javascript was off and
user selected from the userSupportList drop down.
        if(isset($_POST['ajax']['to']) &&
$this->module->getUserId($_POST['ajax']['to'])) {
            $conv->to =
$_POST['ajax']['to'];
            $conv->initiator_id =
$this->
module->getUserId($_POST['ajax']['to']);
        }
        else
            $conv->addError('to','User not
found?');
    }

    if($conv->interlocutor_id && $conv->initiator_id
== $conv->interlocutor_id) {
        $conv->addError('to', "Can't send message to
self!");
    }

    if(!$this->module->isAdmin() && $conv->
interlocutor_id == $this->module->newsUserId){
        $conv->addError('to', "User not found?");
    }

    // check user-to-user perms                if(!$conv->
hasErrors() && !$this->module->
userToUser && !$this->module->isAdmin())
    {
        if(!$this->module->isAdmin($conv->to))
            $conv->addError('to', "Invalid user!");
    }

    $conv->modified = $t;
    $conv->bm_read = Mailbox::INITIATOR_FLAG;
    if($this->module->isAdmin())
        $msg = new Message('admin');
    else
        $msg = new Message('user');
    $msg->text = $_POST['Message']['text'];
    $validate = $conv->
validate(array('text'),false); // html purify
    $msg->created = $t;
    $msg->sender_id = $conv->initiator_id;
    $msg->recipient_id = $conv->interlocutor_id;
    $msg->file_foto =
CUuploadedFile::getInstance($msg, 'file_foto');
    if($this->module->checksums) {
        $msg->crc64 = Message::crc64($msg->text);
    }
    // 64bit INT
    }
    else
        $msg->crc64 = 0;
    // Validate

```

```

        $validate = $conv->validate(null, false); // don't clear errors
        $validate = $msg->validate() && $validate;
if($validate)
    {
        $conv->save();
        $msg->conversation_id = $conv-
>conversation_id;
        $msg->save();
        if(strlen($msg->file_foto)>0)
            $msg->file_foto->saveAs(Yii::app()->basePath
            .'/../images/foto/' . $msg->file_foto);
            Yii::app()->user->setFlash('success',
"Message has been sent!");
            $this->redirect(array('message/inbox'));
        }
        else
        {
            Yii::app()->user->setFlash('error', "Error
sending message!");
        }
    }
    else{
        $conv = new Mailbox();
if(isset($_GET['id']))
        $conv->to = $this->module-
>getUserName($_GET['id']);
        elseif(isset($_GET['to']))
            $conv->to = $_GET['to'];
        else
            $conv->to = '';
            $msg = new Message();
        }
        $this-
>render('compose', array('conv'=>$conv, 'msg'=>$msg));
    }

    public function actionReply()
    {
        if(!$this->module->authManager && (!$this->module>sendMsgs
|| ($this->module->readOnly && !$this->module-
>isAdmin())) )
            $this->redirect(array('message/inbox'));

        $this->module->registerConfig($this->getAction()->getId());

        if($this->module->isAdmin())
            $reply = new Message('admin');
    else
        $reply = new Message('user');

        $conv = Mailbox::conversation($_GET['id']);

        if(isset($_POST['text']))
        {
            $reply->text = $_POST['text'];
            $validate = $conv-

```

```

>validate(array('text'),false); // html_purify $reply-
>conversation_id = $conv-
>conversation_id;
    if($conv->initiator_id != Yii::app()->user->id)
    {
        $reply->recipient_id = $conv-
>initiator_id;
        $conv->bm_read = $conv->bm_read &
~Mailbox::INITIATOR_FLAG;
    }
    else {
        $reply->recipient_id = $conv-
>interlocutor_id;
        $conv->bm_read = $conv->bm_read &
~Mailbox::INTERLOCUTOR_FLAG;
    }

    $reply->sender_id = $this->module->getUserid();

    $reply->created = time();
    $conv->modified = $reply->created;

    $reply->crc64 = Message::crc64($reply->text);

    $conv->bm_deleted = 0; // restore message
    $conv->interlocutor_del = 0;
    $conv->initiator_del = 0;

    $validate = $reply->validate();
    $validate = $conv->validate() && $validate;

    if($validate)
    {
        $conv->save();
        $reply->save();
        Yii::app()->user->setFlash('success',
"Message sent!");
        $this->redirect(array('message/inbox'));
    }
    else{
        Yii::app()->user->setFlash('error', "Error
sending message!");

        $this-
>render('message',array('conv'=>$conv, 'reply'=>$reply));
    }
}

    public function actionView()
    {
        $this->module->registerConfig($this->getAction()->getId());
        $cs = $this->module->getClientScript();
        $cs->registerScriptFile($this->module-
>getAssetsUrl().'js/message.js');
        $js = '$(".mailbox-message-

```

```

list").yiiMailboxMessage('.$this->module->getOptions().");";
        $cs->registerScript('mailbox-
js',$js,CClientScript::POS_READY);

        $conv = Mailbox::conversation($_GET['id']);

        $conv->markRead($this->module->getUserid());
        $reply = new Message;
        $this->render('message',array('conv'=>$conv,
'reply'=>$reply));
    }
}

```

## 8.Users

### a.Model

#### User.php

```

<?php
class User extends CActiveRecord
{
    const STATUS_NOACTIVE=0;          const
    STATUS_ACTIVE=1;
        const STATUS_BANNED=-1;

        //TODO: Delete for next version (backward compatibility)
        const STATUS_BANED=-1;

        /**
         * The followings are the available columns in table 'users':
         * @var integer $id
         * @var string $username
         * @var string $password
         * @var string $email
         * @var string $activkey
         * @var integer $createtime
         * @var integer $lastvisit
         * @var integer $superuser
         * @var integer $status
         * @var timestamp $create_at
         * @var timestamp $lastvisit_at
         */

        /**
         * Returns the static model of the specified AR class.
         * @return CActiveRecord the static model class
         */
        public static function model($className=__CLASS__)
        {
            return parent::model($className);
        }

        /**
         * @return string the associated database table name
         */
        public function tableName()

```

```

    {
        return Yii::app()->getModule('user')->tableUsers;
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.CConsoleApplication
        return ((get_class(Yii::app())=='CConsoleApplication'
        || (get_class(Yii::app())!='CConsoleApplication' && Yii::app()-
        >getModule('user')->isAdmin()))?array(
            array('username', 'length', 'max'=>30, 'min' =>
3,'message' => UserModule::t("Incorrect username (length between 3
and 20 characters).")),
            array('password', 'length', 'max'=>128, 'min' =>
4,'message' => UserModule::t("Incorrect password (minimal length 4
symbols).")),
            array('email', 'email'),
            array('username', 'unique', 'message' =>
UserModule::t("This user's name already exists.")),
            array('email', 'unique', 'message' =>
UserModule::t("This user's email address already exists.")),
            array('username', 'match', 'pattern' => '/^[A-
Za-z0-9_]+$/u','message' => UserModule::t("Incorrect symbols (Az0-
9).")),
            array('status', 'in',
'range'=>array(self::STATUS_NOACTIVE,self::STATUS_ACTIVE,self::STA
TUS_BANNED)),
            array('superuser', 'in', 'range'=>array(0,1)),
array('create_at', 'default', 'value' => date('Y-m-d H:i:s'),
'setOnEmpty' => true, 'on' => 'insert'),
            array('lastvisit_at', 'default', 'value' => '0000-00-
00 00:00:00', 'setOnEmpty' => true, 'on' => 'insert'),
            array('username, email, superuser, status',
'required'),
            array('superuser, status', 'numerical',
'integerOnly'=>true),
            array('id, username, password, email, activkey,
create_at, lastvisit_at, superuser, status', 'safe',
'on'=>'search'),
            ):(Yii::app()->user->id==$this-
>id)?array(
                array('username, email',
'required'),
                array('username', 'length', 'max'=>30, 'min' =>
3,'message' => UserModule::t("Incorrect username (length between 3
and 20 characters).")),
                array('email', 'email'),
                array('username', 'unique', 'message' =>
UserModule::t("This user's name already exists.")),
array('username', 'match', 'pattern' => '/^[AZa-z0-
9_]+$/u','message' => UserModule::t("Incorrect symbols (Az0-
9).")),
                array('email', 'unique', 'message' =>

```

```

UserModule::t("This user's email address already exists."),
):array());
}

/**
 * @return array relational rules.
 */
public function relations()
{
    $relations = Yii::app()->getModule('user')->relations;
    if (!isset($relations['profile']))
        $relations['profile'] = array(self::HAS_ONE,
'Profile', 'user_id');
    return $relations;
}

/**
 * @return array customized attribute labels (name=>label)
 */
public function attributeLabels()
{
    return array(
        'id' => UserModule::t("Id"),
        'username' => UserModule::t("username"),
        'password' => UserModule::t("password"),
        'verifyPassword' => UserModule::t("Retype
Password"),
        'email' => UserModule::t("E-mail"),
        'verifyCode' => UserModule::t("Verification
Code"),
        'activkey' => UserModule::t("activation key"),
        'createtime' => UserModule::t("Registration date"),
        'create_at' => UserModule::t("Registration date"),
        'lastvisit_at' => UserModule::t("Last visit"),
        'superuser' => UserModule::t("Superuser"),
        'status' => UserModule::t("Status"),
    );
}

public function scopes()
{
    return array(
'active' => array(
        'condition' => 'status='.self::STATUS_ACTIVE,
    ),
'notactive' => array(
        'condition' => 'status='.self::STATUS_NOACTIVE,
    ),
'banned' => array(
'condition' => 'status='.self::STATUS_BANNED,
    ),
'superuser' => array(
        'condition' => 'superuser=1',
    ),
'notsafe' => array(

```

```

'select' => 'id, username, password, email, activkey, create_at,
lastvisit_at, superuser, status',
    ),
    );
}

    public function defaultScope()
    {
        return CMap::mergeArray(Yii::app()->getModule('user')-
>defaultScope,array(
            'alias'=>'user',
            'select' => 'user.id, user.username, user.email,
user.create_at, user.lastvisit_at, user.superuser, user.status',
        ));
    }

    public static function itemAlias($type,$code=NULL) {
        $_items = array(
            'UserStatus' => array(
                self::STATUS_NOACTIVE =>
                UserModule::t('Not active'),
                self::STATUS_ACTIVE =>
                UserModule::t('Active'),
                self::STATUS_BANNED =>
                UserModule::t('Banned'),
            ),
            'AdminStatus' => array(
                '0' => UserModule::t('No'),
                '1' => UserModule::t('Yes'),
            ),
        );
        if (isset($code))
            return
            isset($_items[$type][$code]) ?
            $_items[$type][$code] : false;
        else
            return  isset($_items[$type]) ?
            $_items[$type] : false;
    }

/**
 * Retrieves a list of models based on the current
search/filter conditions.
 * @return CActiveDataProvider the data provider that can
return the models based on the search/filter conditions.
 */
    public function search()
    {
        // Warning: Please modify the following code to remove
attributes that
        // should not be searched.
        $criteria=new CDbCriteria;

        $criteria->compare('id',$this->id);
        $criteria->compare('username',$this->username,true);
        $criteria->compare('password',$this->password);
        $criteria->compare('email',$this->email,true);
        $criteria->compare('activkey',$this->activkey);
    }

```

```

$criteria->compare('create_at',$this->create_at);
$criteria->compare('lastvisit_at',$this->lastvisit_at);
$criteria->compare('superuser',$this->superuser);
$criteria->compare('status',$this->status);

return new CActiveDataProvider(get_class($this), array(
    'criteria'=>$criteria,
    'pagination'=>array(
        'pageSize'=>Yii::app()->getModule('user')->user_page_size,
    ),
));
}
public function getCreatetime()
{
    return strtotime($this->create_at);
}
public function setCreatetime($value) {
    $this->create_at=date('Y-m-d H:i:s',$value);
}
public function getLastvisit() {
return strtotime($this->lastvisit_at);
}
public function setLastvisit($value) {
    $this->lastvisit_at=date('Y-m-d H:i:s',$value);
}
public function getFullName() {
return $this->username;
}
public function getSuggest($q) {
    $c = new CDbCriteria();
    $c->addSearchCondition('username', $q, true, 'OR');
$c->addSearchCondition('email', $q, true, 'OR');
return $this->findAll($c);
} }

```

#### **b.View index.php**

```

<?php
$this->breadcrumbs=array(
    UserModule::t('Users')=>array('/user'),
    UserModule::t('Manage'),
);
$this->menu=array(
    array('label'=>UserModule::t('Create User'),
    'url'=>array('create')),
    //array('label'=>UserModule::t('Manage Users'),
    'url'=>array('admin')),
    // array('label'=>UserModule::t('Manage Profile Field'),
    'url'=>array('profileField/admin')),
    //array('label'=>UserModule::t('List User'),
    'url'=>array('/user')),
);
Yii::app()->clientScript->registerScript('search', "
$('.search-button').click(function() {
$('.search-form').toggle();
return
false;
});

```

```

$( '.search-form form' ).submit( function() {
    $.fn.yiiGridView.update( 'user-grid', {
    data: $(this).serialize()
    });
return false;
});
");
?>
<h1><?php echo UserModule::t("Manage Users"); ?></h1>
<p><?php echo UserModule::t("You may optionally enter a comparison
operator (<b>&lt;</b>, <b>&lt;=</b>, <b>&gt;</b>, <b>&gt;=</b>,
<b>&lt;&gt;</b> or <b>=</b>) at the beginning of each of your
search values to specify how the comparison should be done.");
?></p>
<?php $this->widget( 'application.extensions.tablesorter.Sorter',
array(
    'data'=>$records,
    'columns'=>array(
        'username',
        'email',
        'create_at',
        'lastvisit_at',
    ),
)); ?>
profile.ph
p
<?php $this->pageTitle=Yii::app()->name . ' -
'.UserModule::t("Profile");
$this->breadcrumbs=array(
    UserModule::t("Profile"),
);
$this->menu=array(
    ((UserModule::isAdmin())
        ?array('label'=>UserModule::t('Daftar Users'),
'url'=>array('/user/admin'))
        :array()),
    //array('label'=>UserModule::t('List User'),
'url'=>array('/user')),
    array('label'=>UserModule::t('Ubah Data'),
'url'=>array('edit')),
    array('label'=>UserModule::t('Ganti Password'),
'url'=>array('changepassword')),
array('label'=>UserModule::t('Logout'),
'url'=>array('/user/logout')),
);
?>
<h1><?php echo UserModule::t('Your profile'); ?></h1>
<?php if(Yii::app()->user->hasFlash('profileMessage')): ?>
<div class="success">
    <?php echo Yii::app()->user->getFlash('profileMessage'); ?>
</div>
<?php endif; ?>
<table class="dataGrid">
    <tr>
        <th class="label"><?php echo CHtml::encode($model-
>getAttributeLabel('NIM/NIP')); ?></th>

```

```

        <td><?php echo CHtml::encode($model->username); ?></td>
    </tr>
    <?php
        $profileFields=ProfileField::model()->forOwner()-
>sort()->findAll();
        if ($profileFields) {
            foreach($profileFields as $field) {
                //echo "<pre>"; print_r($profile); die();
                ?>
                <tr>
                    <th class="label"><?php echo
                    CHtml::encode(UserModule::t($field->title)); ?></th>
                    <td><?php echo (($field->widgetView($profile))?$field-
                    >widgetView($profile):CHtml::encode(($field-
                    >range)?Profile::range($field->range,$profile-
                    >getAttribute($field->varname)):$profile->getAttribute($field-
                    >varname))); ?></td>
                </tr>
                <?php
            }// $profile->getAttribute($field->varname)
            ?>
            <tr>
                <th class="label"><?php echo CHtml::encode($model-
                >getAttributeLabel('email')); ?></th>
                <td><?php echo CHtml::encode($model->email); ?></td>
            </tr>
            <tr>
                <th
                    class="label"><?php
                    echo
                    CHtml::encode($model->getAttributeLabel('Tanggal Daftar')); ?></th>
                <td><?php echo $model->create_at; ?></td>
            </tr>
            <tr>
                <th
                    class="label"><?php
                    echo
                    CHtml::encode($model->getAttributeLabel('Terakhir Masuk')); ?></th>
                <td><?php echo $model->lastvisit_at; ?></td>
            </tr>
            <tr>
                <th
                    class="label"><?php
                    echo
                    CHtml::encode($model->getAttributeLabel('Status')); ?></th>
                <td><?php echo
                    CHtml::encode(User::itemAlias("UserStatus",$model->status));
                    ?></td>
            </tr>
        </table>

```

### c. Controller

#### UserController.php

```

<?php
class UserController extends Controller
{
    /**
     * @var CActiveRecord the currently loaded data model instance.
     */
    private $_model;

```

```

    /**
 * @return array action filters
 */
    public function filters()
    {
        return CMap::mergeArray(parent::filters(), array(
            //'rights',
            'accessControl', // perform access control for
CRUD operations
            'postOnly + delete', // we only allow deletion via POST request
        ));
    }
    /**
 * Specifies the access control rules.
 * This method is used by the 'accessControl' filter.
 * @return array access control rules
 */
    public function accessRules()
    {
        return array(
            array('allow', // allow admin user to perform
'admin' and 'delete' actions

            'actions'=>array('admin','delete','create','update','view'),
                'hak
akses'=>array('SuperAdmin','Staff','Kaprodi'),
            ),
            array('deny', // deny all users
                'users'=>array('*'),
            ),
        );
    }

    /**
 * Displays a particular model.
 */
    public function actionView()
    {
        $model = $this->loadModel();
        $this->render('view', array(
            'model'=>$model,
        ));
    }

    /**
 * Lists all models.
 */
    public function actionIndex()
    {
        $dataProvider=new CActiveDataProvider('User', array(
            'criteria'=>array(
                'condition'=>'status>.User::STATUS_BANNED,
            ),
            'pagination'=>array(

```

```

        'pageSize'=>Yii::app()->controller-
>module->user_page_size,
        ),
    ));

    $this->render('index',array(
        'dataProvider'=>$dataProvider,
    ));
}

public function actionUpdate()
{
    $model=$this->loadModel();
    $profile=$model->profile;
    $this->performAjaxValidation(array($model,$profile));
    if(isset($_POST['User']))
    {
        $model->attributes=$_POST['User'];
        $profile->attributes=$_POST['Profile'];

        if($model->validate()&&$profile->validate())
        {
            $old_password = User::model()->notsafe()-
>findByPk($model->id);
            if ($old_password->password!=$model-
>password) {
                $model->password=Yii::app()-
>controller->module->encrypting($model->password);
                $model->activkey=Yii::app()-
>controller->module->encrypting(microtime().$model->password);
            }
            $model->save();
            $profile->save();
            $this->redirect(array('view','id'=>$model-
>id));
        } else $profile->validate();
    }

    $this->render('update',array(
        'model'=>$model,
        'profile'=>$profile,
    ));
}

public function actionDelete()
{
    if(Yii::app()->request->isPostRequest)
    {
        // we only allow deletion via POST request
        $model = $this->loadModel();
        $profile = Profile::model()->findByPk($model-
>id);

        $profile->delete();
        $model->delete();
        // if AJAX request (triggered by deletion via
admin grid view), we should not redirect the browser
        if(!isset($_POST['ajax']))

```

```

        $this->redirect(array('/user/admin'));
    }
    else
        throw new
CHttpException(400,'Invalid request.
Please do not repeat this request again.');
```

```

    }

    protected function performAjaxValidation($validate)
    {
        if(isset($_POST['ajax']) && $_POST['ajax']=== 'user-form')
        {
            echo CActiveForm::validate($validate);
            Yii::app()->end();
        }
    }
}
/**
 * Returns the data model based on the primary key given in
the GET variable.
 * If the data model is not found, an HTTP exception will be
raised.
 */
public function loadModel()
{
    if($this->_model===null)
    {
        if(isset($_GET['id']))
            $this->_model=User::model()->notsafe()-
>findByPk($_GET['id']);
        if($this->_model===null)
            throw new CHttpException(404,'The
requested page does not exist.');
```

```

    }
    return $this->_model;
}
/**
 * Returns the data model based on the primary key given in the
GET variable.
 * If the data model is not found, an HTTP exception will be
raised.
 * @param integer the primary key value. Defaults to null,
meaning using the 'id' GET variable
 */
public function loadUser($id=null)
{
    if($this->_model===null)
    {
        if($id!==null || isset($_GET['id']))
            $this->_model=User::model()-
>findByPk($id!==null ? $id : $_GET['id']);
        if($this->_model===null)
            throw
new CHttpException(404,'The
requested page does not exist.');
```

```

    }
    return $this->_model;
}
}
}

```

