```php
<?php

/*! \mainpage Artikkelihallintaohjelma
 *
 * \section intro_sec Introduction
 *
 * Tämän on artikkelienhallintaohjelmiston lahdekoodin dokumentoiti.
 *
 * \section outro_sec Author
 * Daniel Tisza, 2011
 * Jussi Isosävi, 2012
 *
 */

/**
 * Copyright: @author Daniel Tisza, 2011
 */

session_start();
srand( time() ); // Seed random generator
                 // Force HTTPS, not implemented

/*!
* \class Root
*/
class Root {
}

/*!
* \class Input
* \brief     syöttökenttä
*
* \details    lomakkeella yksi syötekenttä
*
*/

class Input    extends Root
{
  var $name;
  var $type;
  var $value;
  var $size;
  var $errormsg;

/*!
 * \brief     luomisifunktio
 * \details    luodaan uusi syötekenttä
 *
 * \param    name        käyttäjälle näytettävä nimi
 * \param    type        käyttäjälle näytettävä tyyppi
 *                        (numero/teksti)
 * \param    value        käyttälle näytettävä oletusarvo
 * \param    size        kentän koko
 * \param    errormsg    käyttäjälle näytettävä virheteksti,
 *                        jos arvo ei kelpaa
 */
  function Input( $name, $type, $value, $size, $errormsg )
  {
    $this->name = $name;
    $this->type = $type;
    $this->value = $value;
    $this->size = $size;
    $this->errormsg = $errormsg;
  }
```

```
            /*!
            * \brief          Luo HTML Input tägin.
            * \details        Luo muuttujien arvoilla html-tägin.
            *
            * \return         palauttaa html-tägin tekstinä.
            */
             function html( )
             {
               $html = '<input name="'.md5( $this->name ).'"'.
                           ' type="'.$this->type.'"'.
                           ' size="'.$this->size.'"';

               if( strcmp( $this->type, 'file' ) == 0 )
               {
                 $html .= '>'.$this->errormsg;
               }
               else
               {
                 $html .= ' value="'.$this->value.'">';

                 if ( $this->value == '' )
                 {
                   // ilmoitetaan virheviesti jos kenttä on tyhjä
                   $html .= $this->errormsg;
                 }
               }

               return $html;
             }

            /*!
            * \brief         muotoilee syötekentän tekstiksi.
            * \brief         Tekstissä on ensin nimi ja arvo
            *                kaksoispisteellä erotettuna.
            *
            *  \return     palauttaa syötekentän tekstinä
            */
             function txt( )
             {
               if( strcmp( $this->type, 'submit' ) == 0 ||
                   strcmp( $this->type, 'file' ) == 0 )
               {
                 return;
               }

               $txt = $this->name.' : '.$this->value;

               return $txt;
             }


            /*!
            * \brief         Tiedosto-syötekentän tiedoston nimi.
            * \details        Tiedosto-tyyppisessä syötekentässä ladatun tiedoston nimi.
            *
            * \return         palauttaa väliaikaisen tiedostonimen.
            */
             function filename( )
             {
               $filename = '';

               if( strcmp( $this->type, 'file' ) == 0 )
               {
                 $filename = $_FILES[ md5( $this->name ) ]['tmp_name'];
               }

               return $filename;
             }
```

```
 /*!
 * \brief         Kerää tiedot lomakkeen syötekentistä
 * \details     Kerää tiedot lomakkeen syötekentistä
 *
 *  \return     Palautetaan 1, jos kentän arvo kelpaa.
 *                muussa tapauksessa 0.
 */

  function verified_fill( )
  {
    if( strcmp( $this->type, 'submit' ) == 0 )
      return;

    if( strcmp( $this->type, 'file' ) == 0 )
    {
      $this->value = $_FILES[ md5( $this->name ) ]['name'];
    }
    else
    {
      $post = $_POST[ md5( $this->name ) ];
      $post = trim( $post );
      $post = substr( $post, 0, $this->size );
      $post = escapeshellcmd( $post );

      $this->value = $post;
    }

    if ( $this->value == '' ) //* jos arvo on tyhjä */
    {
      return 0; //* huono arvo */
    }

    return 1; //* arvo kelpaa */
  }
}


/*!
 * \class Form
 * \brief         Tietojensyöttölomake
 * \details     kerää tiedot lomakkeen syötekentistä
 *
 * \details        lomake sisältää syöttökenttiä ja tunnistekentän
 *
 */
class Form extends Root
{
  var $action;
  var $method;
  var $enctype;
  var $inputs;
  var $captcha_input;
  var $submit_input;
  var $captcha_img_url;
```

```
 /*!
 * \brief     luomisifunktio
 * \details    luodaan uusi lomake
 *
 * \param    action               Osoite, johon lomake lähetetään
 * \param    method               Tapa, jolla lomake lähetettän (GET/POST)
 * \param    enctype               muoto, jossa lomake lähetetään palvelimelle
 *                               (multipart/form-data)
 * \param    captcha_img_url       Osoite, mistä tunnistekuva haetaan
 *
*/
  function Form( $action, $method, $enctype, $captcha_img_url )
  {
    $this->action = $action;
    $this->method = $method;
    $this->enctype = $enctype;
    $this->inputs = array( );
    $this->captcha_input = new Input( 'Verification Code in Image', '', '', 5 );
    $this->submit_input = new Input( '', 'submit', 'Submit', 10 );
    $this->captcha_img_url = $captcha_img_url;
  }

 /*!
 * \brief     Syötekentän lisäys-funktio
 * \details    Lisätään syötekenttä lomekkeelle
 *          edellisten perään taulukkoon
 *
 * \param    input    Lisättävä syötekentä
 *
*/
  function add_input( $input )
  {
    $this->inputs[] = $input;
  }

/*!
 * \brief     Lomakkeen täytön tarkistaminen.
 * \details    Käydään lläpi kaikki lomakkeen tiedot.
 *
 * \return  Hyväksytyn täytön tunnisteessa
 */
  function verified_fill( )
  {
    $valid_fill = 1;

    foreach( $this->inputs as $key => $val )
    {
      if ( $val->verified_fill( ) != 1 )
      {
        // Joku kenttä on väärin -> lomake väärin
        $valid_fill = 0;
      }
    }

    return $valid_fill;
  }

/*!
 * \brief         Muodosta lomakkeen aloitustägi.
 * \details        Muodosta lomakkeen aloitustägi.
 *
 * \return         Palauttaa html-tägin tekstinä.
 */
  function begin_html( )
  {
    $html = '<form action="'.$this->action.'"'.
                  ' method="'.$this->method.'"'.
                  ' enctype="'.$this->enctype.'">';

    return $html;
  }
```

```
  /*!
  * \brief         Muodosta lomakkeen lopetustägi.
  * \details        Lisää tunnistekuvan ja lopetustägin.
  *
  * \return         Palauttaa html-tägin tekstinä.
  */
   function end_html( )
   {
     $html = '<img src="'.$this->captcha_img_url.'"><br>'.
             $this->captcha_input->name.' '.$this->captcha_input->html( ).'<br>'.
             $this->submit_input->html( ).
             '</form>';

     return $html;
   }

  /*!
  * \brief             Muodota syötekenttien tägit.
  * \details            Syötekentät laitetaan taulukkoon.
  *
  * \return             syötekenttien tägit tekstinä.
  */
   function inputs_html( )
   {
     $html = '<table>';

     foreach ( $this->inputs as $key => $val )
     {
       $html .= '<tr><td>'.$val->name.'</td><td>'.$val->html( ).'</td></tr>';
     }

     $html .= '</table>';

     return $html;
   }

 /*!
  * \brief         Kerää lomakkeen tiedot tesktinä.
  * \details        Kerää lomakkeen tiedot tesktinä ja
  *              jokainen syötekenttä omalla rivillään.
  *
  * \return         Palauttaa lomakkeen tiedot tekstinä.
  */
   function inputs_txt( )
   {
     $txt = '';

     foreach ( $this->inputs as $key => $val )
     {
       $txt .= $val->txt( )."\r\n";
     }

     return $txt;
   }

 /*!
  * \brief         Kerää lomakkeen Tiedosto-syötekenttien tiedostonimet.
  * \details     Tiedosto-tyyppisessä syötekentässä ladatun tiedoston nimi.
  *
  * \return         palauttaa taulukon väliaikeisista tiedostonimistä.
  */
   function inputs_filenames( )
   {
     $filenames = array( );

     foreach ( $this->inputs as $key => $val )
     {
       $filename = $val->filename( );

       if ( $filename != '' )
       {
         $filenames[] = array('tmp_name' => $filename, 'name' => $val->value );
       }
     }

     return $filenames;
   }
```

```
/*!
 * \brief     Tunnisteen tarkistus-funktio
 * \details    Tarkistaa tunnisteen oikeellisuuden.
 *
 * \return  Palauttaa 1, jos tunniste on oikein
 *          ja muissa tapauksissa 0.
 */

  function verify_captcha( )
  {
    $this->captcha_input->verified_fill( );
    $entered_captcha = $this->captcha_input->value;
    $this->captcha_input->value = '';

    if ( $entered_captcha != '' )
    {
      $captchahash = md5( $entered_captcha.$_SESSION['captchasalt'] );

      if ( strcmp( $_SESSION['captchahash'], $captchahash ) == 0 )
      {
        unset( $_SESSION['captchahash'] );
        unset( $_SESSION['captchasalt'] );
        return 1;
      }
    }

    unset( $_SESSION['captchahash'] );
    unset( $_SESSION['captchasalt'] );
    return 0;
  }
}

/**
 * @brief     Tunnistekuvassa olevan tekstin generointi
 * @return     Arvottu tunnisteteksti
 */

function captcha_text( )
{
  /* määritellään tunnisteen arvot */
  $chars = array( 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'm',
                  'n', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'y', 'z',
                  '2', '3', '4', '5', '6', '8', '9', '@', '#', '%', '?',
                  'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'J', 'K', 'L', 'M',
                  'N', 'P', 'R', 'S', 'T', 'U', 'V', 'W', 'Y', 'Z' );
  $text = '';

  /* arvojen arpominen */
  for ( $i = 0; $i < 5; $i++ )
  {
    $text .= $chars[ rand( 0, count( $chars ) - 1 ) ];
  }

  return $text; /* palautetaan saatu arvo */
}
```

```
 /**
 * @brief     Tulostaa sivulle Tunnistekuvan.
 */
function captcha_image( )
{
  header( 'Content-type: image/jpeg' );

  $text = captcha_text( );
  $salt = rand();
  $_SESSION['captchasalt'] = $salt;
  $_SESSION['captchahash'] = md5( $text.$salt );   // Salted hash of solution

  /* määritettään tunnisteen leveys,korkeus, */
  $w = 30;
  $h = 30;
  $charw = 2.3 * $w;
  $charh = 2.3 * $h;
  $zw = 400;
  $zh = 150;

  $imgzoom = imagecreatetruecolor( $zw, $zh );
  $zoomwhite = imagecolorallocate( $imgzoom, 255, 255, 255 );

  for ( $i = 0; $i < 5; $i++ ) // Taustan värit,kirjaimien väri
  {
    $img = imagecreatetruecolor( $w, $h );
    $color = imagecolorallocate( $img, 255, 255, 255 );

    imagechar( $img, 5, 0, 0, $text[ $i ], $color );
    $imgrot = imagerotate( $img, rand( -30, 30 ), 0 );
    imagedestroy( $img );

    $x = 8 + rand( 60, 65 ) * $i;
    $y = 5 + rand( 0, 70 );

    imagecopyresampled( $imgzoom, $imgrot, $x, $y, 0, 0, $charw, $charh, $w, $h );
    imagedestroy( $imgrot );
  }

  for ( $i = 0; $i < 5; $i++ ) // Taustan viivat
  {
    imageline( $imgzoom, rand( 0, $zw ), rand( 0, $zh ),
                         rand( 0, $zw ), rand( 0, $zh ), $zoomwhite );
  }

  for ( $i = 0; $i < 3; $i++ ) // Taustan kaaret
  {
    imagearc( $imgzoom, rand( 0, $zw ), rand( 0, $zh ),
                        rand( 0, $zw ), rand( 0, $zh ),
                        rand( 0, 360 ), rand( 0, 360 ), $zoomwhite );
  }

  imagejpeg( $imgzoom, NULL, 15 );
  imagedestroy( $imgzoom );
}
```

```
        /**
         * @brief     Muotoilee ja lähettää sähköpostin.
         *
         * @param     from     lähettäjän osoite
         * @param     to         vastaanottajan osoite
         * @param     subject    aihe
         * @param     msg     viesti
         *
         * \return Palauttaa lähetys-funktion paluuarvon.
         */

        function format_send_email( $from, $to, $subject, $msg )
        {
          $headers = "From: $from\n".
                     "Reply-To: $from\n".
                     "Return-Path: $from\n".
                     "MIME-Version: 1.0\n".
                     "Content-type: text/plain; charset=iso-8859-1\n";

          $msg = wordwrap( $msg, 69, "\r\n" );

          $res = mail( $to, $subject, $msg, $headers );

          return $res;
        }

        /**
         * @brief     Muotoilee ja lähettää sähköpostin liitetiedoston kanssa.
         *
         * @param     from                lähettäjän osoite
         * @param     to                    vastaanottajan osoite
         * @param     subject               aihe
         * @param     filename            tiedostonimi
         * @param     attachedfilename     liitetiedoston nimi
         *
         * \return Palauttaa lähetys-funktion paluuarvon.
         */
        function send_email_file( $from, $to, $subject, $filename, $attachedfilename )
        {
          $headers = "From: $from\n".
                     "Reply-To: $from\n".
                     "Return-Path: $from\n".
                     "MIME-Version: 1.0\n".
                     'Content-type: application/zip; name="'.$attachedfilename.'"'."\n".
                     "Content-Transfer-Encoding: base64\n".
                     "Content-Disposition: attachment\n";

          $msg = chunk_split(base64_encode( file_get_contents( $filename ) ) );

          $res = mail( $to, $subject, $msg, $headers );

          return $res;
        }

        /**
         * @brief     Turha funktio?
         *
         * @param     originalname     ?
         * @param     tmpname         ?
         */
        function add_extension( $originalname, $tmpname )
        {
        }
```

```
/**
 * @brief     pakkaa annetun tiedoston.
 *
 * @param     filename     pakattavan tiedoston nimi.
 *
 * \return     paluttaa pakatun tiedoston nimen.
 */
function temporary_zip_file( $filename )
{
  $tmpdir = '/tmp/';
  $random = md5( date( 'r', time() ) );
  $tmpzip = $tmpdir.$random.'.zip';

  exec( 'zip '.$tmpzip.' '.$filename ); //luo .zip-tiedoston

  $f = fopen( $tmpzip, 'r' ); //todentaa .zip-tiedoston

  if ( $f )
  {
    fclose( $f );
  }
  else
  {
    return 0;
  }

  return $tmpzip;
}

if ( $_GET['captcha'] == '1' )
{
  captcha_image( );
  exit;
}

$captcha_img_url = $_SERVER['SCRIPT_NAME'].'?captcha=1&nocache='.md5( rand() );

$form = new Form( $_SERVER['SCRIPT_NAME'], 'post',
                  'multipart/form-data', $captcha_img_url );

$form->add_input( new Input( 'First Name', '', '', 20, 'Must be non-empty' ) );
$form->add_input( new Input( 'Last Name', '', '', 20, 'Must be non-empty' ) );
$form->add_input( new Input( 'Email Address', '', '', 20, 'Must be non-empty' ) );
$form->add_input( new Input( 'Institution', '', '', 40, 'Must be non-empty' ) );
$form->add_input( new Input( 'Publication Title', '', '', 40, 'Must be non-empty' ) );
$form->add_input( new Input( 'Publication File', 'file', '', 30, 'Must be non-empty' ) );

echo '<html>';
```

```
if ( $form->verified_fill( ) && $form->verify_captcha( ) )
{

  $from = 'k83660@student.uwasa.fi'; /* mistä lähetään*/
  $to = 'jussiisosavi@gmail.com'; /* minne lähetetään*/

  $res = format_send_email( $from, $to,  'Information', $form->inputs_txt( ) );

  if ( $res )
  {
    echo 'Information Submitted.<br>';
  }
  else
  {
    echo 'Information Submission Failed.<br>';
  }

  foreach ( $form->inputs_filenames( ) as $key => $val )
  {

    $file_extension = pathinfo( $val['name'], PATHINFO_EXTENSION );

    $filename = $val['tmp_name'].'.'.$file_extension;

    if ( rename( $val['tmp_name'], $filename ) != true )
    {
      echo 'Unable to use file extension. File not submitted.<br>';
      continue;
    }

    $tmpzip = temporary_zip_file( $filename );

    if ( $tmpzip === 0 )
    {
      echo 'Unable to compress file.';
      continue;
    }

    $res = send_email_file( $from, $to, 'Publication File',
                            $tmpzip, 'Publication.zip' );

    if ( $res )
    {
      echo 'File Submitted.<br>';
    }
    else
    {
      echo 'File Not Submitted.<br>';
    }
  }

}
else
{
  echo $form->begin_html( ),
       $form->inputs_html( ),
       $form->end_html( );
}

echo '</html>';

?>
```