Dynamische Websites

Week 9

AGENDA

- **Nut van een framework?**
- Relatieve URLs
- Views
  - Slicing
  - 2 step design

NUT VAN EEN FRAMEWORK?

- Heel veel code is voor jullie al geschreven, jullie moeten deze op de juiste plaatsen kunnen gebruiken en aanroepen.
  - System folder kan je in verschillende projecten herbruiken
  - Application folder daar komt de specifieke functionaliteit van je project telkens in

Agenda

- Nut van een framework?
- **Relatieve URLs**
- Views
  - Slicing
  - 2 step design

CATEGORY_DETAIL.PHP

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Dynamische Websites</title>
    <link href="/2013/lessons/theorie08_template/frontcontroller_uitbreiding_category/application/css/bootstrap.c
    <link href="/2013/lessons/theorie08_template/frontcontroller_uitbreiding_category/application/css/eigenstijl.cs
    <div class="navbar navbar-inverse navbar-fixed-top">
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="#">Vehicles example</a>
            </div>
            <div class="navbar-collapse collapse">
                <ul class="nav navbar-nav">
                    <li class=""><a href="/2013/lessons/theorie08_template/frontcontroller_uitbreiding_category/home
                    <li class=""><a href="/2013/lessons/theorie08_template/frontcontroller_uitbreiding_category/home
                    <li class=""><a href="/2013/lessons/theorie08_template/frontcontroller_uitbreiding_category/vehic
                    <li class=""><a href="/2013/lessons/theorie08_template/frontcontroller_uitbreiding_category/categ
                </ul>
            </div>
        </div>
    </div>
</head>
```

VEHICLE_DETAIL.PHP

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Dynamische Websites</title>
    <link href="/2013/lessons/theorie08_template/frontcontroller_uitbreiding_category/application/css/bootstrap.c
    <link href="/2013/lessons/theorie08_template/frontcontroller_uitbreiding_category/application/css/eigenstijl.cs
    <div class="navbar navbar-inverse navbar-fixed-top">
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="#">Vehicles example</a>
            </div>
            <div class="navbar-collapse collapse">
                <ul class="nav navbar-nav">
                    <li class=""><a href="/2013/lessons/theorie08_template/frontcontroller_uitbreiding_category/home
                    <li class=""><a href="/2013/lessons/theorie08_template/frontcontroller_uitbreiding_category/home
                    <li class=""><a href="/2013/lessons/theorie08_template/frontcontroller_uitbreiding_category/vehic
                    <li class=""><a href="/2013/lessons/theorie08_template/frontcontroller_uitbreiding_category/categ
                </ul>
            </div>
        </div>
    </div>
</head>
```

URLs

- Wat als het path nu verandert?
  - Overal aanpassen???

CATEGORY_DETAIL.PHP

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Dynamische Websites</title>
    <link href="<?php echo baseUrl('/application/css/bootstrap.css'); ?>" rel="stylesheet" />
    <link href="<?php echo baseUrl('/application/css/eigenstijl.css'); ?>" rel="stylesheet" />
    <div class="navbar navbar-inverse navbar-fixed-top">
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="#">Vehicles example</a>
            </div>
            <div class="navbar-collapse collapse">
                <ul class="nav navbar-nav">
                    <li class=""><a href="<?php echo baseUrl('home'); ?>">Home</a></li>
                    <li class=""><a href="<?php echo baseUrl('home/display/dynweb'); ?>">Display</a></li>
                    <li class=""><a href="<?php echo baseUrl('vehicle/index'); ?>">Vehicle</a></li>
                    <li class=""><a href="<?php echo baseUrl('category/index'); ?>">Category</a></li>
                </ul>
            </div>
        </div>
    </div>
</head>
```

VIEWHELPERS.PHP

```php
<?php

function baseUrl($uri = '')
{
    return str_replace('index.php', '', $_SERVER['SCRIPT_NAME']) . $uri;
}

...
```

$_SERVER['SCRIPT_NAME'] =
/2013/lessons/theorie08_template/frontcontroller_uitbreiding_category
$uri =
/application/css/bootstrap.css

return geeft
/2013/lessons/theorie08_template/
frontcontroller_uitbreiding_category/application/css/bootstrap.css

NAVBAR.PHP

```
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-
collapse">
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" href="#">Vehicles Example</a>
        </div>
        <div class="navbar-collapse collapse">
            <ul class="nav navbar-nav">
                <?php foreach ($this->menuItems as $menuitem): ?>
                    <li class="<?php echo ($menuitem['link'] == getCurrentPath()) ? 'active' : ''; ?>"><a
href="<?php echo baseUrl($menuitem['link']); ?>"><?php echo $menuitem['description']; ?></a></li>
                <?php endforeach; ?>
            </ul>
        </div>
    </div>
</div>
```

VIEWHELPERS.PHP

```
<?php

function getCurrentPath()
{
    $path =  substr($_SERVER['REQUEST_URI'],
strlen(dirname($_SERVER['SCRIPT_NAME'])) + 1);
    $path = trim($path, '/');

    // if no scriptfile in de request_uri, we assume index.php is called
    $path = ($path) ? $path : 'index.php';

    return $path;
}



...
```

HTML VAN NAVBAR

```
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            <button type="button" class="navbar-toggle" data-toggle="collapse"
data-target=".navbar-collapse">
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" href="#">Vehicles Example</a>
        </div>
        <div class="navbar-collapse collapse">
            <ul class="nav navbar-nav">
              <li class="active"><a href="/2013/lessons/theorie08_template/
2stepdesign_stap03/home">home</a></li>
              <li class=""><a href="/2013/lessons/theorie08_template/
2stepdesign_stap03/home/display/dynweb">display</a></li>
              <li class=""><a href="/2013/lessons/theorie08_template/
2stepdesign_stap03/vehicle">vehicles</a></li>
              <li class=""><a href="/2013/lessons/theorie08_template/
2stepdesign_stap03/category">categories</a></li>
            </ul>
        </div>
    </div>
</div>
```

RELATIEVE URLS

- Zie
  - dynweb2013examples/svn/theorie09/RelatieveURLs/frontcontroller_uitbreiding_relatieve_URLs

Agenda

- Nut van een framework?
- Relatieve URLs
- **Views**
  - Slicing
  - 2 step design

VIEWS

- In een view staat alles
  - header, footer, ...
- Tot nu toe hebben we de views via require_once binnen gehaald of getoond

VOORBEELD

- Footer in Vehicles voorbeeld

HOME.PHP

```
<!DOCTYPE html>
<html lang="en">
<head>
...
</head>
<body>
...
<footer>
    <p>questions?
        <a href="mailto:elke.steegmans@khleuven.be">Elke Steegmans</a> or
        <a href="mailto:kurt.beheydt@khleuven.be">Kurt Beheydt</a>
    </p>
</footer>

</div>
</body>
</html>
```

DISPLAY.PHP

```
<!DOCTYPE html>
<html lang="en">
<head>
...
</head>
<body>
...
<footer>
    <p>questions?
        <a href="mailto:elke.steegmans@khleuven.be">Elke Steegmans</a> or
        <a href="mailto:kurt.beheydt@khleuven.be">Kurt Beheydt</a>
    </p>
</footer>

</div>
</body>
</html>
```

VEHICLE_OVERVIEW.PHP

```
<!DOCTYPE html>
<html lang="en">
<head>
...
</head>
<body>
...
<footer>
    <p>questions?
        <a href="mailto:elke.steegmans@khleuven.be">Elke Steegmans</a> or
        <a href="mailto:kurt.beheydt@khleuven.be">Kurt Beheydt</a>
    </p>
</footer>

</div>
</body>
</html>
```

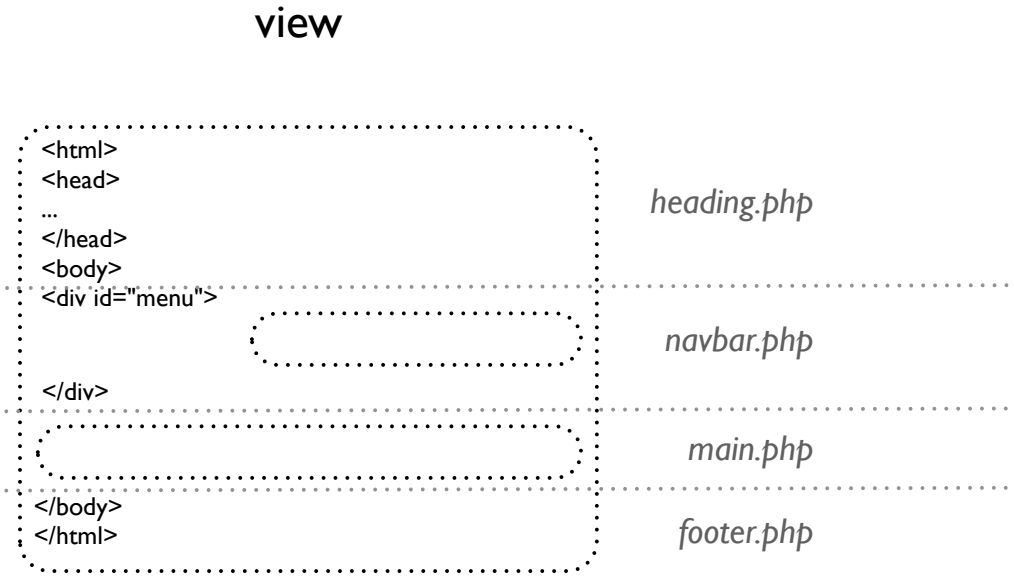
VOORBEELD

- Uitbreiding
  - We zijn vergeten om mevrouw Kemme ook toe te voegen als contactpersoon
- Probleem
  - In alle view files moeten we nu de footer aanpassen
    - heel veel dubbele code

Agenda

- Nut van een framework?
- Relatieve URLs
- **Views**
  - **Slicing**
  - 2 step design

SLICING


view
<html>
<head>
...
</head>
<body>

</body>
</html>
heading.php
navbar.php
main.php
footer.php

OK?

SIMPEL VOORBEELD

- Simpele home pagina met een menubar, header, footer en inhoud

- Zie
  - dynweb2013examples/svn/theorie09/2stepdesign/slicing_simple_example

HOME_VIEW.PHP

```
<?php require_once('layout_header.php'); ?>

<?php require_once('layout_navbar.php'); ?>

    <h1>sliced theme</h1>

    <div class="row">
        <div class="span12">
            <h3>Welcome to this sliced example.</h3>
        </div>
    </div>
    <hr>

<?php require_once('layout_footer.php'); ?>
```

LAYOUT_HEADER.PHP

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title><?php echo $title; ?></title>
    <link href="css/bootstrap.css" rel="stylesheet" />
    <link href="css/eigenstijl.css" rel="stylesheet" />
</head>
<body>
```

LAYOUT_NAVBAR.PHP

```
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" href="#">Slicing Example</a>
        </div>
        <div class="navbar-collapse collapse">
            <ul class="nav navbar-nav">
                <li class=""><a href="#">Home</a></li>
                <li class=""><a href="#">Item 1</a></li>
                <li class=""><a href="#">Item 2</a></li>
                <li class=""><a href="#">Item 3</a></li>
            </ul>
        </div>
    </div>
</div>
```

** **

LAYOUT_FOOTER.PHP

```
  <footer>
    <p>questions?
      <a href="mailto:elke.steegmans@khleuven.be">
      Elke Steegmans</a> or
       <a href="mailto:mieke.kemme@khleuven.be">
      Mieke Kemme</a> or
      <a href="mailto:kurt.beheydt@khleuven.be">
      Kurt Beheydt</a>
      </p>
  </footer>
</div>
</body>
</html>
```

SLICING

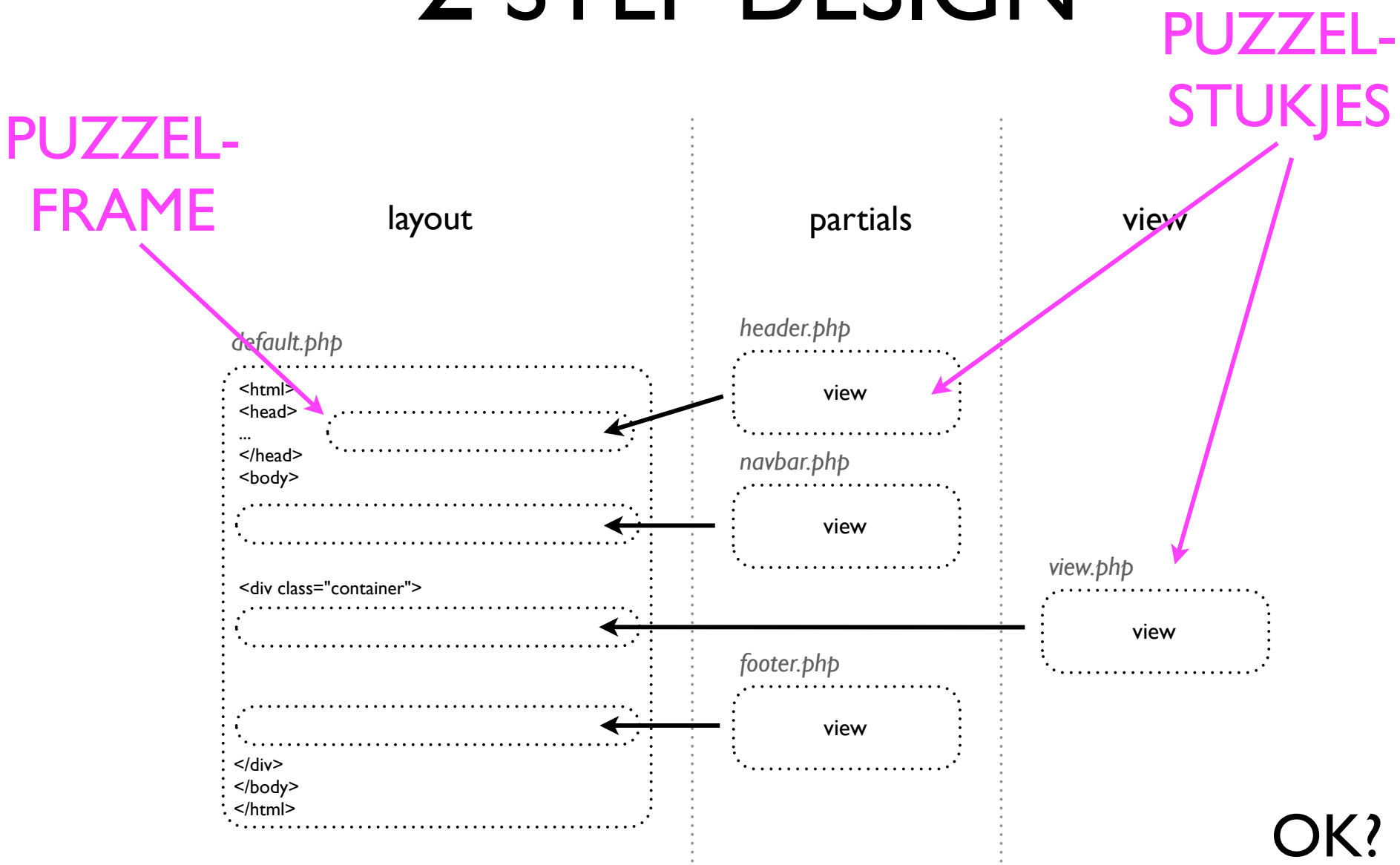
- Probleem
  - indien we de layout_navbar.php eruit laten
    - invalid HTML-pagina
    - pagina's op zich zijn niet herbruikbaar

NOK!

AGENDA

- Front controller
- Relatieve URLs
- **Views**
  - Slicing
  - **2 step design**

2 STEP DESIGN



OK?

2 STEP DESIGN

- Stap 1
  - logische view opbouwen
  - puzzelframe en alle puzzelstukjes klaar leggen
- Stap 2
  - HTML genereren
  - op basis van de puzzel (= puzzelframe en puzzelstukjes) nu de HTML genereren

2 STEP DESIGN - STAP 1 SIMPEL VOORBEELD

- Simpele home pagina met een menubar, header, footer en inhoud

- Zie
  - dynweb2013examples/svn/theorie09/2stepdesign/2stepdesign_simple_example

DEFAULT.PHP

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
<?php require_once('partial_header.php'); ?>
<?php require_once('partial_navbar.php'); ?>
</head>
<body>


<div class="container">

<?php require_once('home_view.php'); ?>

<?php require_once('partial_footer.php'); ?>

</div>
</body>
</html>
```

PARTIAL_HEADER.PHP

```
<title><?php echo $title; ?></title>
<link href="css/bootstrap.css" rel="stylesheet" />
<link href="css/eigenstijl.css" rel="stylesheet" />
```

PARTIAL_NAVBAR.PHP

```
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-co
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" href="#">Slicing Example</a>
        </div>
        <div class="navbar-collapse collapse">
            <ul class="nav navbar-nav">
                <li class=""><a href="#">Home</a></li>
                <li class=""><a href="#">Item 1</a></li>
                <li class=""><a href="#">Item 2</a></li>
                <li class=""><a href="#">Item 3</a></li>
            </ul>
        </div>
    </div>
</div>
```

HOME_VIEW.PHP

```
<h1>two step design pattern</h1>

<p>Welcome to this two step design pattern example.</p>
```

PARTIAL_FOOTER.PHP

```
<footer>
    <p>questions?
     <a href="mailto:elke.steegmans@khleuven.be">Elke Steegmans</a>
     or
      <a href="mailto:mieke.kemme@khleuven.be">Mieke Kemme</a>
     or
      <a href="mailto:kurt.beheydt@khleuven.be">Kurt Beheydt</a>
    </p>
</footer>
```

2 STEP DESIGN - STAP 1
SIMPEL VOORBEELD

- Elk stukje van de view is een op zich staand stuk code en kan herbruikt worden in andere voorbeelden
  - probleem van Slicing is weggewerkt hierdoor

OK!

2 STEP DESIGN - STAP 1 VEHICLE VOORBEELD

- Op elke pagina zijn de volgende HTML stukken niet hetzelfde
  - het grootste gedeelte van de body
  - de header waar de titel verschillend is
  - de navbar waar de active URL telkens verschillend is

2 STEP DESIGN - STAP 1 VEHICLE VOORBEELD

- Voorbeeld
  - u0034562.webontwerp.khleuven.be/theorie09/2stepdesign/2stepdesign_stap00/vehicle/index

- Zie
  - dynweb2013examples/svn/theorie09/2stepdesign/2stepdesign_stap00

INDEX.PHP

```
<?php
session_start();

define('APPLICATION_PATH', 'application/');
define('SYSTEM_PATH', 'system/');

require_once(APPLICATION_PATH . 'config.php');

require_once(SYSTEM_PATH . 'model/Db.php');
require_once(SYSTEM_PATH . 'controller/Controller.php');
require_once(SYSTEM_PATH . 'view/viewHelpers.php');

$controller = new Controller();
$controller->run();
```

Code Onveranderd

CONTROLLER.PHP

```
<?php

class Controller
{
    private function parseUri() {
        $scriptprefix = str_replace(self::CONTROLLER_FILE, '', $_SERVER['SCRIPT_NAME']);
        $uri = str_replace(self::CONTROLLER_FILE, '', $_SERVER['REQUEST_URI']);

        $path = substr($uri, strlen($scriptprefix));
        $path = preg_replace('/[^a-zA-Z0-9]\//', "", $path);
        $path = trim($path, '/');

        @list($controller, $action, $params) = explode("/", $path, 3);

        if (isset($controller)) {
            $this->setController($controller);
        }
        if (isset($action)) {
            $this->setAction($action);
        }
        if (isset($params)) {
            $this->setParams(explode("/", $params));
        }
    }
```

Code Onveranderd

CONTROLLER.PHP

```
...
    public function run()
    {
        // checking the parameter count, using Reflection (http://www.php.net/reflection)
        $reflector = new ReflectionClass($this->controller);
        $method = $reflector->getMethod($this->action);
        $parameters = $method->getNumberOfRequiredParameters();

        if (($parameters) > count($this->params)) {
            die("Action '$this->action' in class '$this->controller' expects $parameters mandatory
parameter(s), you only provided " . count($this->params) . ".");
        }

        // create an instance of the controller as an object
        $controller = new $this->controller();

        // call the method based on $this->action and the params
        call_user_func_array(array($controller, $this->action), $this->params);
    }

}
```

Code Onveranderd

VehicleController.php

```
<?php
require_once(SYSTEM_PATH . 'view/Template.php');

class VehicleController
{

    private $_vehicleMapper;
    private $_template;

    private $_view;
    private $_pageTitle;
    private $_menuItems;
    private $_text;

    public function __construct ()
    {
        require_once(APPLICATION_PATH . 'model/VehicleMapper.php');

        $this->_vehicleMapper = new VehicleMapper();
    }
```

VehicleController.php

```
...

    public function index()
    {
        $this->_view = 'vehicle_overview.php';
        $this->_pageTitle = 'Vehicles';

        $menuMapper = new MenuMapper();
        $this->_menuItems = $menuMapper->getMenuItems();

        $this->_object = $this->_vehicleMapper->getAll();

        require_once(APPLICATION_PATH . 'view/default.php');
    }

...

}
```

MENUMAPPER.PHP

```php
<?php

class MenuMapper
{

    public function getMenuItems()
    {
        $menuitems = array(
            array(
                'link' => 'home',
                'description' => 'home',
            ),
            array(
                'link' => 'home/display/dynweb',
                'description' => 'display',
            ),
            array(
                'link' => 'vehicle',
                'description' => 'vehicles',
            ),
        );

        return $menuitems;
    }
}
```

VehicleMapper.php

```
<?php
require_once(SYSTEM_PATH . 'model/Db.php');
require_once(APPLICATION_PATH . 'model/Vehicle.php');

class VehicleMapper
{
    private $_db;

    public function __construct()
    {
        $this->_db = Db::getInstance();
    }

    public function add($object) {...}

    public function getAll() {
        $query = "
            SELECT *
            FROM vehicles
        ";
        return $this->_db->queryOne($query, 'Vehicle');
     }

    public function get($id) {...}

}
```

Code Onveranderd

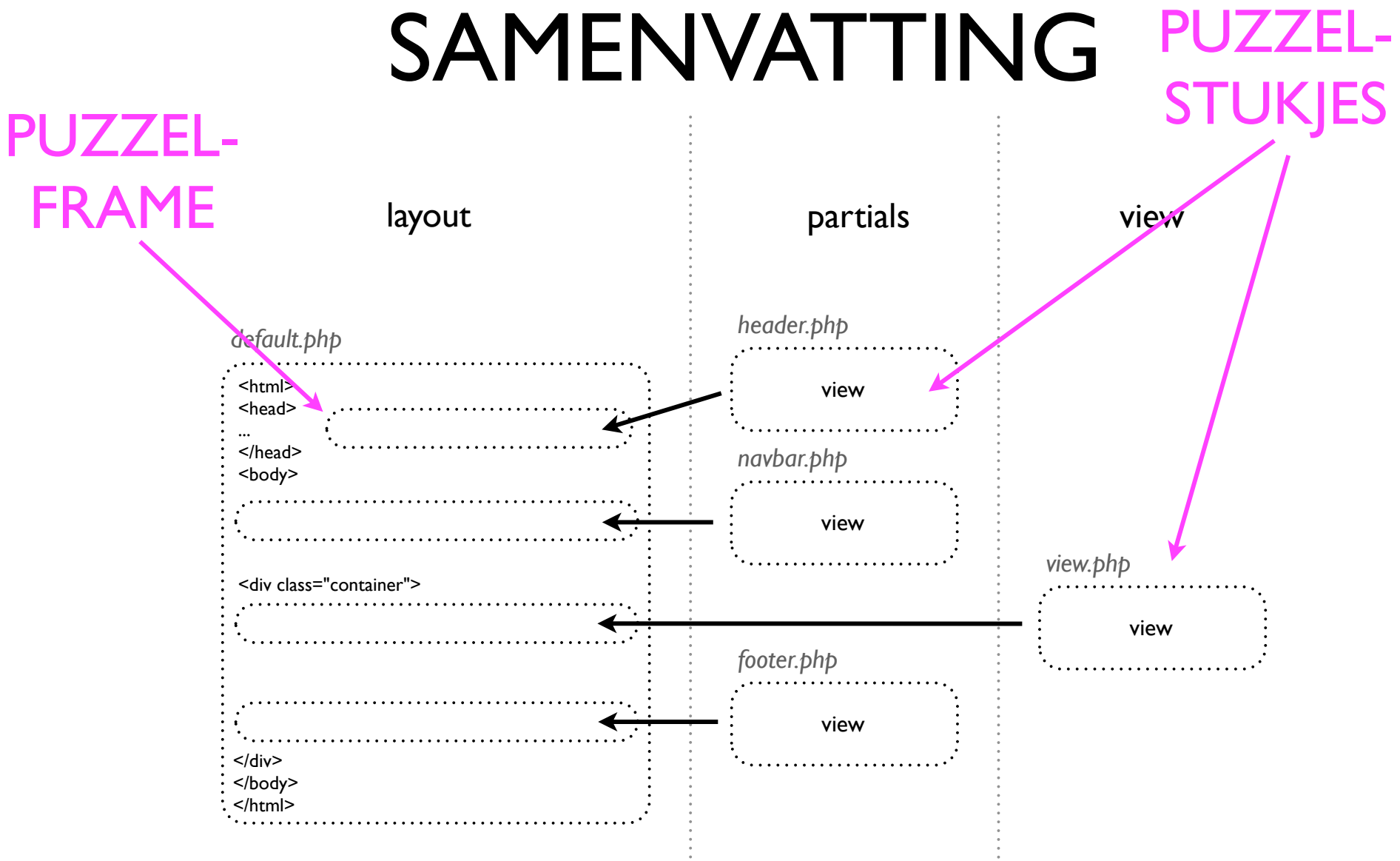
DEFAULT.PHP

```
<!DOCTYPE html>
<html lang="en">
<head>
    <?php require_once('headermeta.php'); ?>
</head>

<body>

<?php require_once('navbar.php'); ?>

<div class="container">
    <h1><?php echo $this->_pageTitle; ?></h1>

    <?php require_once("$this->_view"); ?>
    <hr>

    <?php require_once('footer.php'); ?>
</div>

</body>
</html>
```

HEADERMETA.PHP

```
<meta charset="utf-8">
<title><?php echo $this->_pageTitle; ?></title>
<link href="<?php echo baseUrl('css/bootstrap.css'); ?>" rel="stylesheet">
<link href="<?php echo baseUrl('css/eigenstijl.css'); ?>" rel="stylesheet">
```

2 STEP DESIGN - STAP 1 SAMENVATTING



2 STEP DESIGN REFACTORING STAP 1

- Probleem 1
  - Stel dat we de default.php file anders willen noemen
    - => op heel veel plaatsen nu code aanpassen
  - We willen een klasse die we in andere projecten ook kunnen gebruiken om views mee op te bouwen

2 STEP DESIGN REFACTORING STAP I

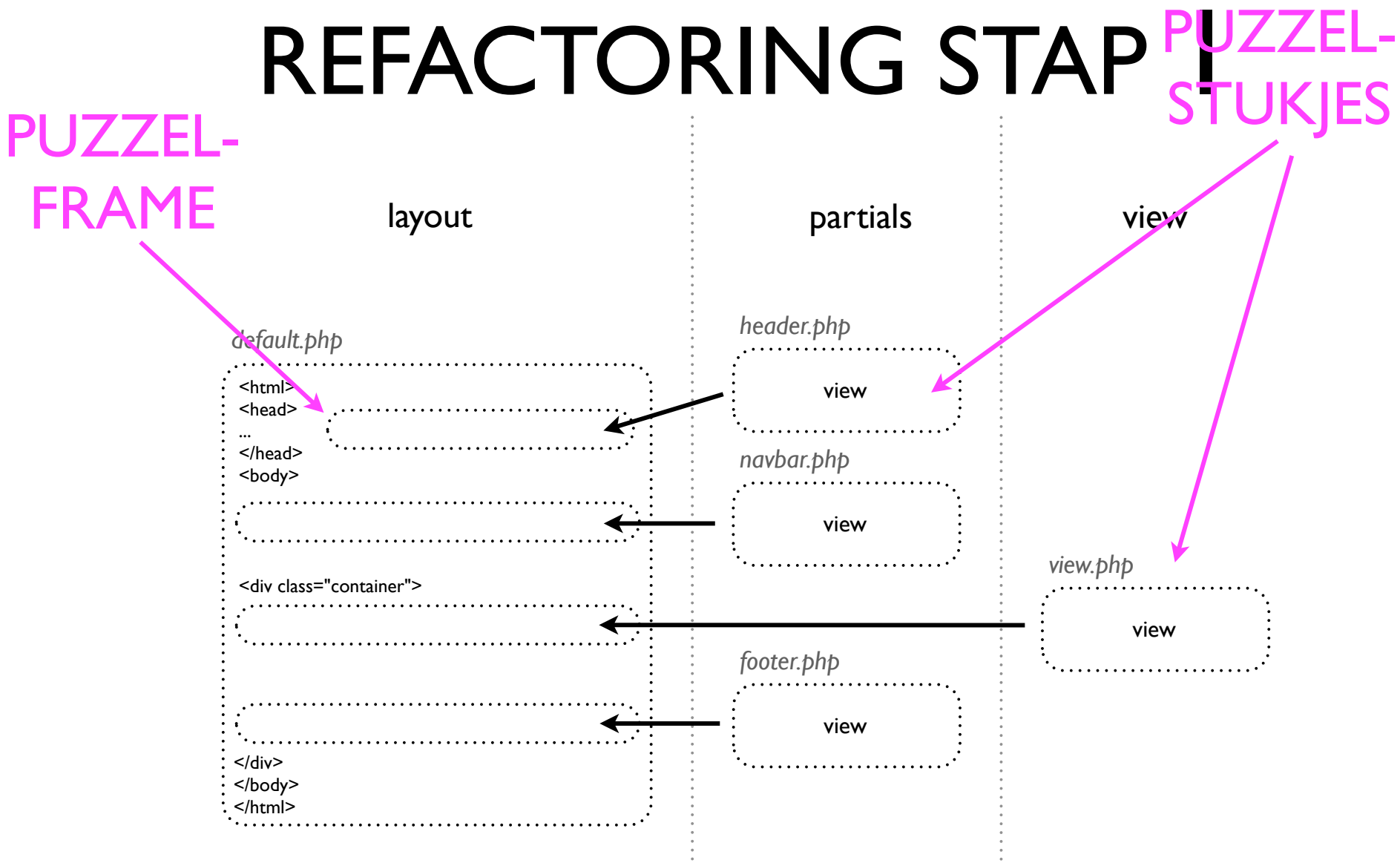
- Probleem 2
  - Eigenlijk ook geen 2 step design op dit moment
    - je neemt je puzzel, dan begin je al HTML te maken, dan leg je het eerste puzzelstukje erin, en maak je verder HTML, en zo ga je verder voor ieder puzzelstukje

2 STEP DESIGN REFACTORING STAP I

- Oplossing
  - Template klasse
    - hier de 2 stappen in implementeren en herbruikbaarheid
      - eerst partials en view zetten (eerst puzzelframe en puzzelstukjes maken)
      - daarna renderen van volledige HTML (daarna HTML renderen van volledige puzzel (= puzzelframe + puzzelstukjes))

2 STEP DESIGN

REFACTORING STAP I



2 STEP DESIGN - STAP 1+2 VEHICLE VOORBEELD

- Toon het overzicht van alle vehicles die tot nu toe in de DB zitten
  - vehicle/index
- Zie
  - dynweb2013examples/svn/theorie09/2stepdesign_stap01

INDEX.PHP

```
<?php
session_start();

define('APPLICATION_PATH', 'application/');
define('SYSTEM_PATH', 'system/');

require_once(APPLICATION_PATH . 'config.php');

require_once(SYSTEM_PATH . 'model/Db.php');
require_once(SYSTEM_PATH . 'controller/Controller.php');
require_once(SYSTEM_PATH . 'view/viewHelpers.php');

$controller = new Controller();
$controller->run();
```

Code Onveranderd

CONTROLLER.PHP

```
<?php

class Controller
{
    private function parseUri() {
        $scriptprefix = str_replace(self::CONTROLLER_FILE, '', $_SERVER['SCRIPT_NAME']);
        $uri = str_replace(self::CONTROLLER_FILE, '', $_SERVER['REQUEST_URI']);

        $path = substr($uri, strlen($scriptprefix));
        $path = preg_replace('/[^a-zA-Z0-9]\//', "", $path);
        $path = trim($path, '/');

        @list($controller, $action, $params) = explode("/", $path, 3);

        if (isset($controller)) {
            $this->setController($controller);
        }
        if (isset($action)) {
            $this->setAction($action);
        }
        if (isset($params)) {
            $this->setParams(explode("/", $params));
        }
    }
```

Code Onveranderd

CONTROLLER.PHP

```
...
    public function run()
    {
        // checking the parameter count, using Reflection (http://www.php.net/reflection)
        $reflector = new ReflectionClass($this->controller);
        $method = $reflector->getMethod($this->action);
        $parameters = $method->getNumberOfRequiredParameters();

        if (($parameters) > count($this->params)) {
            die("Action '$this->action' in class '$this->controller' expects $parameters mandatory
parameter(s), you only provided " . count($this->params) . ".");
        }

        // create an instance of the controller as an object
        $controller = new $this->controller();

        // call the method based on $this->action and the params
        call_user_func_array(array($controller, $this->action), $this->params);
    }

}
```

Code Onveranderd

VehicleController.php

```
<?php
require_once(SYSTEM_PATH . 'view/Template.php');
require_once(APPLICATION_PATH . 'model/MenuMapper.php');

class VehicleController
{

    private $_vehicleMapper;
    private $_template;

    public function __construct ()
    {
        $this->_template = new Template();

        $menuMapper = new MenuMapper();
        $this->_template->menuItems = $menuMapper->getMenuItems();

        $this->_template->setPartial('navbar');
        $this->_template->setPartial('headermeta');
        $this->_template->setPartial('footer');

        require_once(APPLICATION_PATH . 'model/VehicleMapper.php');
        $this->_vehicleMapper = new VehicleMapper();
    }
```

TEMPLATE.PHP

```php
<?php
class Template {

    const TEMPLATE_PATH = 'application/view/';
    const LAYOUT_FOLDER = 'layouts/';
    const PARTIALS_FOLDER = 'partials/';

    protected $data = array();
    protected $content = array();
    protected $partials = array();
    protected $layoutfile;
    private $pagetitle;

    // on instantiation: check the layoutfile
    public function __construct($layoutfile = 'default')
    {
        $layoutfile = self::LAYOUT_FOLDER . $this->addExtension($layoutfile);

        if (file_exists(self::TEMPLATE_PATH . $layoutfile)) {
            $this->layoutfile = self::TEMPLATE_PATH . $layoutfile;
        } else {
            error_log("not found: $layoutfile");
            exit;
        }
    }
```

$this->layoutfile = application/view/layouts/default.php

VehicleController.php

```
<?php
require_once(SYSTEM_PATH . 'view/Template.php');
require_once(APPLICATION_PATH . 'model/MenuMapper.php');

class VehicleController
{

    private $_vehicleMapper;
    private $_template;

    public function __construct ()
    {
        $this->_template = new Template();

        $menuMapper = new MenuMapper();
        $this->_template->menuItems = $menuMapper->getMenuItems();

        $this->_template->setPartial('navbar');
        $this->_template->setPartial('headermeta');
        $this->_template->setPartial('footer');

        require_once(APPLICATION_PATH . 'model/VehicleMapper.php');
        $this->_vehicleMapper = new VehicleMapper();
    }
```

MENUMAPPER.PHP

```php
<?php
class MenuMapper
{

    public function getMenuItems()
    {
        $menuitems = array(
            array(
                'link' => 'home',
                'description' => 'home',
            ),
            array(
                'link' => 'home/display/dynweb',
                'description' => 'display',
            ),
            array(
                'link' => 'vehicle',
                'description' => 'vehicles',
            ),
        );

        return $menuitems;
    }
}
```

VehicleController.php

```
<?php
require_once(SYSTEM_PATH . 'view/Template.php');
require_once(APPLICATION_PATH . 'model/MenuMapper.php');

class VehicleController
{

    private $_vehicleMapper;
    private $_template;

    public function __construct ()
    {
        $this->_template = new Template();

        $menuMapper = new MenuMapper();
        $this->_template->menuItems = $menuMapper->getMenuItems();

        $this->_template->setPartial('navbar');
        $this->_template->setPartial('headermeta');
        $this->_template->setPartial('footer');

        require_once(APPLICATION_PATH . 'model/VehicleMapper.php');
        $this->_vehicleMapper = new VehicleMapper();
    }
```

TEMPLATE.PHP

```
<?php
class Template {

...

// automatic getter and setter, remapping every value to the protected attribute
 // read more about this on www.php.net/manual/en/language.oop5.overloading.php
 public function __set($name, $value)
 {
     $this->data[$name] = $value;
 }
```

$this->data['menuItems'] =
array die getMenuItems methode
van MenuMapper klasse teruggeeft

VehicleController.php

```
<?php
require_once(SYSTEM_PATH . 'view/Template.php');
require_once(APPLICATION_PATH . 'model/MenuMapper.php');

class VehicleController
{

    private $_vehicleMapper;
    private $_template;

    public function __construct ()
    {
        $this->_template = new Template();

        $menuMapper = new MenuMapper();
        $this->_template->menuItems = $menuMapper->getMenuItems();

        $this->_template->setPartial('navbar');
        $this->_template->setPartial('headermeta')
        $this->_template->setPartial('footer');

        require_once(APPLICATION_PATH . 'model/VehicleMapper.php');
        $this->_vehicleMapper = new VehicleMapper();
    }
```

TEMPLATE.PHP

```
// helper to generate a partial
 public function setPartial($partialname, $partialfile = '')
 {
     // if $partialfile is not set, use the partialname as filename
     $partialfile = ($partialfile) ? $partialfile : $partialname;

     $partialfile = self::TEMPLATE_PATH . self::PARTIALS_FOLDER . $this->addExtension($partialfile);

     if (file_exists($partialfile)) {
         $this->partials[$partialname] = $partialfile;
     } else {
         error_log("not found: $partialfile");
         exit;
     }

     return $this;
 }
```

```
$this->partials = array {
        'navbar' => 'application/view/partials/navbar.php'
        'headermeta' => 'application/view/partials/headermeta.php'
        'footer' => 'application/view/partials/footer.php' }
```

VehicleController.php

```
...

    public function index()
    {
        $this->_template->_vehicles = $this->_vehicleMapper->getAll();

        $this->_template->setPagetitle('Vehicles');

        $this->_template->render('vehicle_overview');
    }

...

}
```

TEMPLATE.PHP

```
public function setPagetitle($title)
{
    $this->pagetitle = $title;
}
```

$this->pagetitle = ‘Vehicles’

VehicleController.php

```
...

    public function index()
    {
        $this->_template->_object = $this->_vehicleMapper->getAll();

        $this->_template->setPagetitle('Vehicles');

        $this->_template->render('vehicle_overview');
    }

...

}
```

TEMPLATE.PHP

```
// render the main content of the site
 // while rendering the layout, render the partials as well
 public function render($templatefile)
 {
     $templatefile = $this->addExtension($templatefile);

     if (file_exists(self::TEMPLATE_PATH . $templatefile)) {
         $this->content = $this->renderView(self::TEMPLATE_PATH . $templatefile);
     } else {
         error_log("not found: $templatefile");
         exit;
     }

     $this->renderLayout();
 }
```

$this->content = volledige HTML van vehicle_overview als een string

TEMPLATE.PHP

```
private function renderView($view)
{
    ob_start();
    include($view);
    $data = ob_get_contents();
    ob_end_clean();

    return $data;
}
```

deze functie maakt een buffer in waar char per char inkomt en dus waar uiteindelijk 1 grote string in zit
(de view die we meegeven als parameter)

TEMPLATE.PHP

```
// render the main content of the site
 // while rendering the layout, render the partials as well
 public function render($templatefile)
 {
     $templatefile = $this->addExtension($templatefile);

     if (file_exists(self::TEMPLATE_PATH . $templatefile)) {
         $this->content = $this->renderView(self::TEMPLATE_PATH . $templatefile);
     }
     else {
         error_log("not found: $templatefile");
         exit;
     }

     $this->renderLayout();
 }
```

TEMPLATE.PHP

```
public function renderLayout()
{
    include($this->layoutfile);
}
```

$this->layoutfile =
application/view/layouts/default.php

dit haalt default.php binnen en gaat
dit lijn per lijn uitvoeren

DEFAULT.PHP

```
<!DOCTYPE html>
<html lang="en">
<head>
    <?php $this->renderPartial('headermeta'); ?>
</head>

<body>

<?php $this->renderPartial('navbar'); ?>

<div class="container">
    <h1><?php echo $this->getPagetitle(); ?></h1>

    <?php $this->getContent(); ?>
    <hr>

    <?php $this->renderPartial('footer'); ?>
</div>

</body>
</html>
```

TEMPLATE.PHP

```
public function renderPartial($name)
  {
      if (array_key_exists($name, $this->partials)) {
          echo $this->renderView($this->partials[$name]);
      } else {
          error_log("partial not rendered: $name");
      }
  }
```

print de inhoud van headermeta, dan van navbar, ...

2 STEP DESIGN - STAP 1+2 SAMENVATTING

- Template klasse
  - stap 1: eerst wordt het puzzelframe en de puzzelstukjes gezet
    - puzzelframe: default.php
    - puzzelstukjes: header.php, navbar.php, view.php en footer.php
  - stap 2: dan wordt de HTML pas gerenderd (dus de HTML wordt in 1 keer gerenderd als de puzzel klaar is)

2 STEP DESIGN - STAP 1+2 UITBREIDING

- Uitbreiding
  - zorg er ook voor dat je het overzicht van alle categorieen kunt zien

- Zie
  - dynweb2013examples/svn/ theorie09/2stepdesign_stap02_uitbreiding_category

2 STEP DESIGN - STAP 1+2 NOG VERDERE REFACTORING

- Probleem
  - veel dubbele code
    - refactoren door super klasse van Controller klassen te maken

- Zie
  - dynweb2013examples/svn/theorie09/2stepdesign_stap03

2 STEP DESIGN - STAP 1+2 NOG VERDERE REFACTORING

- Oplossing
  - Controller klasse
    - noemt nu FrontController klasse
  - Superklasse van HomeController, VehicleController, ... gemaakt
    - noemt Controller klasse

FRONTCONTROLLER.PHP

```php
<?php

class FrontController
{
    private function parseUri() {
        $scriptprefix = str_replace(self::CONTROLLER_FILE, '', $_SERVER['SCRIPT_NAME']);
        $uri = str_replace(self::CONTROLLER_FILE, '', $_SERVER['REQUEST_URI']);

        $path = substr($uri, strlen($scriptprefix));
        $path = preg_replace('/[^a-zA-Z0-9]\//', "", $path);
        $path = trim($path, '/');

        @list($controller, $action, $params) = explode("/", $path, 3);

        if (isset($controller)) {
            $this->setController($controller);
        }
        if (isset($action)) {
            $this->setAction($action);
        }
        if (isset($params)) {
            $this->setParams(explode("/", $params));
        }
    }
```

FRONTCONTROLLER.PHP

```
...
    public function run()
    {
        // checking the parameter count, using Reflection (http://www.php.net/reflection)
        $reflector = new ReflectionClass($this->controller);
        $method = $reflector->getMethod($this->action);
        $parameters = $method->getNumberOfRequiredParameters();

        if (($parameters) > count($this->params)) {
            die("Action '$this->action' in class '$this->controller' expects $parameters mandatory
parameter(s), you only provided " . count($this->params) . ".");
        }

        // create an instance of the controller as an object
        $controller = new $this->controller();

        // call the method based on $this->action and the params
        call_user_func_array(array($controller, $this->action), $this->params);
    }

}
```

CONTROLLER.PHP

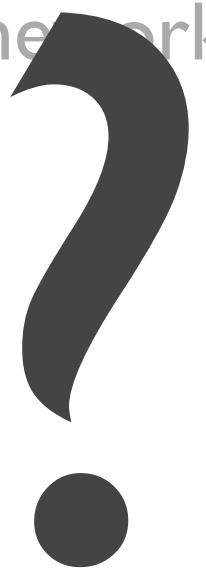
```
<?php
require_once(SYSTEM_PATH . 'view/Template.php');
require_once(APPLICATION_PATH . 'model/MenuMapper.php');

class Controller
{
    protected $_template;

    public function __construct ()
    {
        $this->_template = new Template();

        $menuMapper = new MenuMapper();
        $this->_template->menuItems = $menuMapper->getMenuItems();

        // method chaining in php. http://www.php.net/manual/en/language.references.return.php#78123
        $this->_template->setPartial('navbar')
            ->setPartial('headermeta')
            ->setPartial('footer');
    }
}
```

HOMECONTROLLER.PHP

```
<?php
require_once(SYSTEM_PATH . 'controller/Controller.php');

class HomeController extends Controller
{

    public function __construct () {
        parent::__construct();
    }

    public function index()
    {
        $this->_template->setPagetitle('Welcome to controller home, action index');
        $this->_template->render('home');
    }

    public function display($text = 'no text')
    {
        $this->_template->setPagetitle('Welcome to controller home, action display');
        $this->_template->text = $text;
        $this->_template->render('home_display');
    }
}
```

Agenda

- Nut van een framework?
- Relatieve URLs
- Views
  - Slicing
  - 2 step design



LABO

- Kopieer de laatste versie van de system folder naar je eigen project
- Refactor de rest van je project (je application folder) zodat je geen dubbele code in je views hebt en werk volgens het 2 step design patroon
  - begin met player_overview