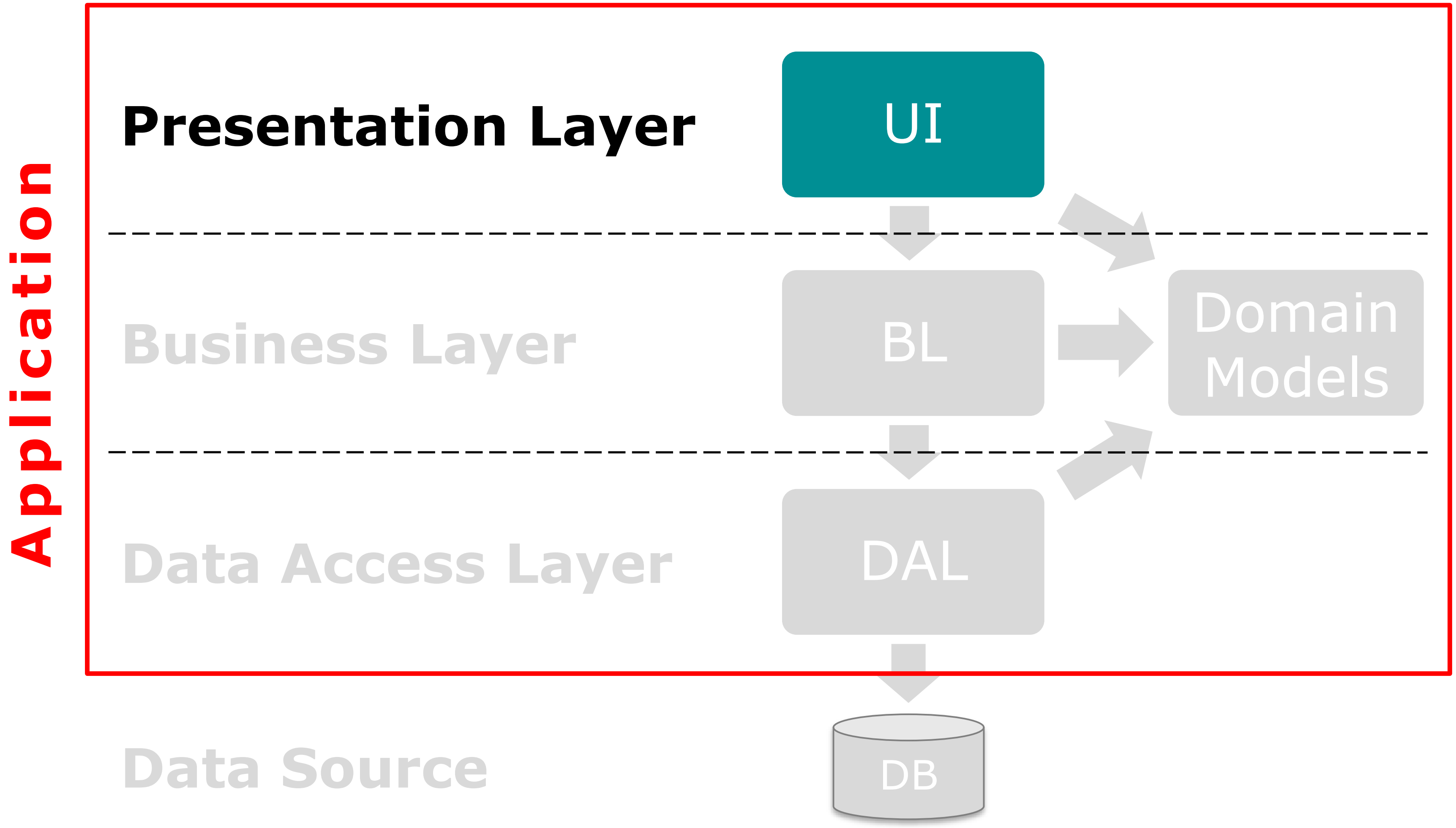
Example 'SupportCenter'

ASP.NET MVC

Info

- De applicatie moet voorzien worden van een web interface
- Hiervoor gaan we gebruik maken van het ASP.NET MVC-framework

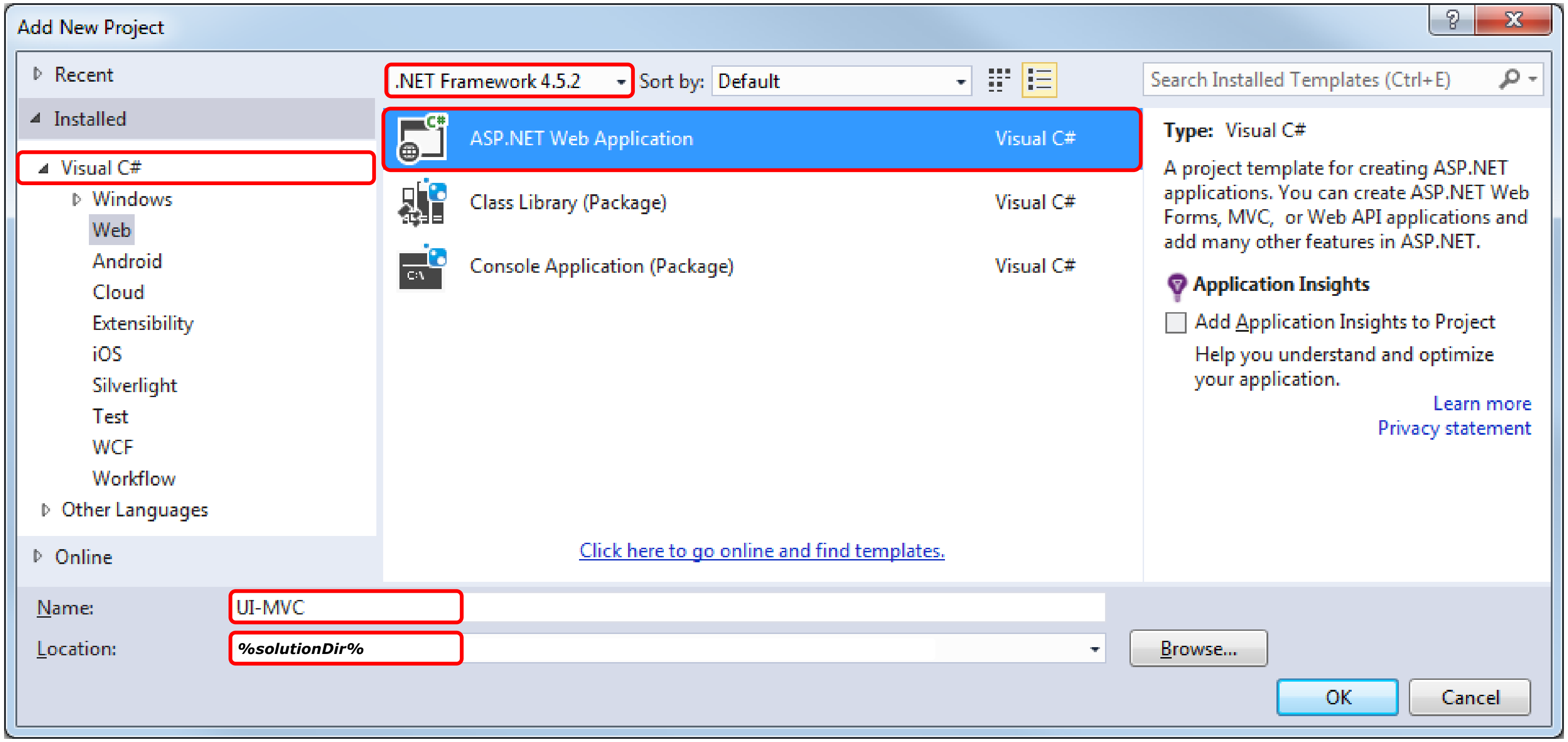
N-Tier architectuur



Oefening

- Voeg een nieuw project 'UI-MVC' toe
  - Type: ASP.NET Web Application
    - Technologie: MVC
    - Authentication: 'No Authentication'
    - Windows Azure: DON'T host in the cloud

project 'UI-MVC'



Add New Project

.NET Framework 4.5.2

ASP.NET Web Application	Visual C#
Class Library (Package)	Visual C#
Console Application (Package)	Visual C#

Click here to go online and find templates.

Type: Visual C#

A project template for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.

Application Insights

☐ Add Application Insights to Project

Help you understand and optimize your application.

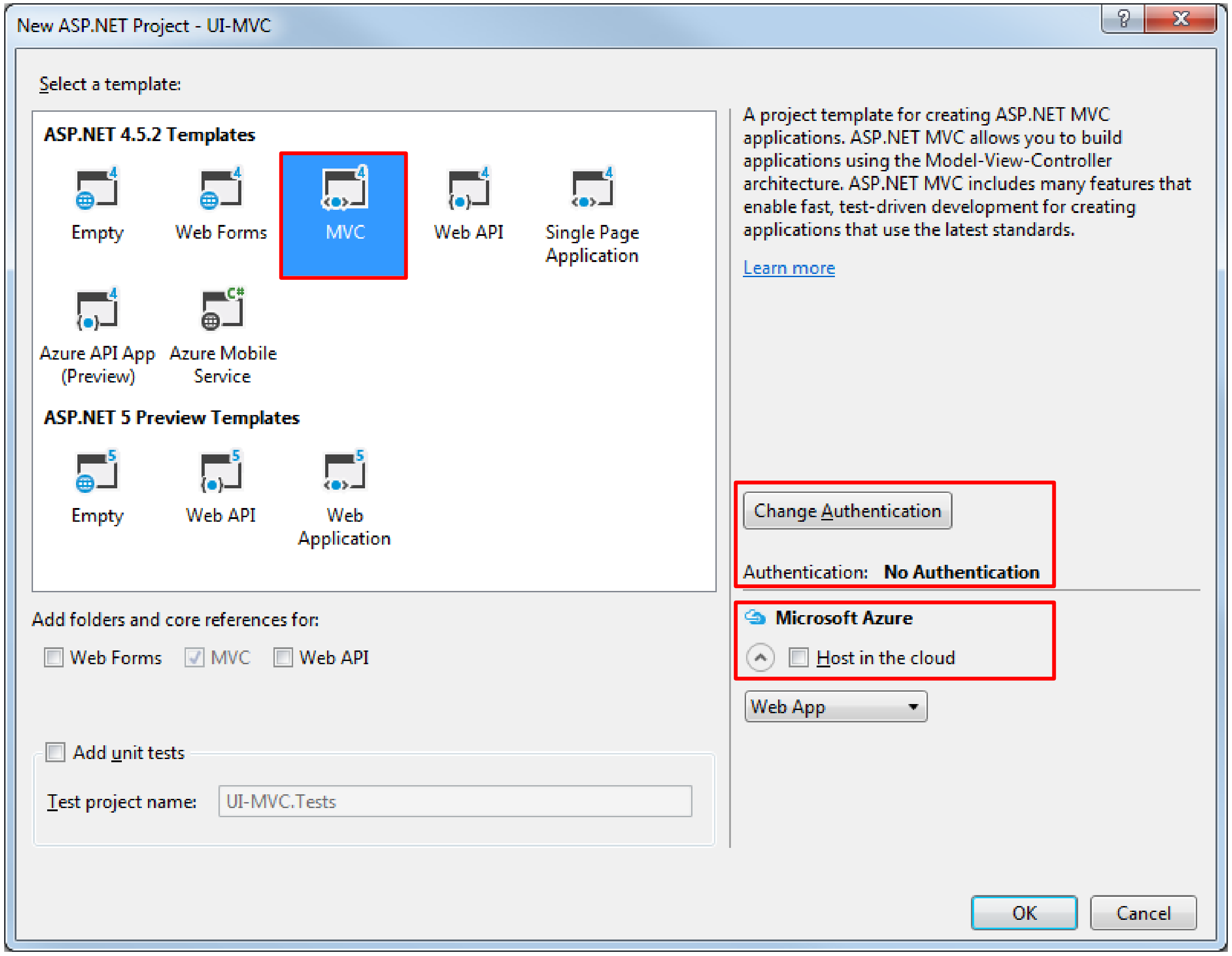
Learn more
Privacy statement

Name: UI-MVC

Location: *%solutionDir%*

Karel de Grote Hogeschool

project 'UI-MVC'


New ASP.NET Project - UI-MVC

Select a template:

ASP.NET 4.5.2 Templates

ASP.NET 5 Preview Templates

A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

Learn more

Change Authentication

Authentication: **No Authentication**

Microsoft Azure

☐ Host in the cloud

Web App

Add folders and core references for:

☐ Web Forms ☑ MVC ☐ Web API

☐ Add unit tests

Test project name: UI-MVC.Tests

Oefening

- Project 'UI-MVC'
  - StartUp-project!!
  - References:
    - project 'BL'
    - project 'Domain'

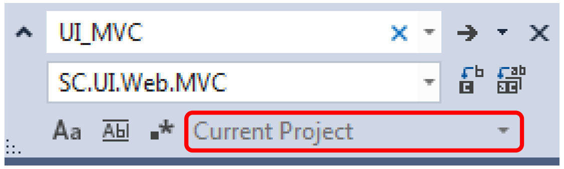
Oefening

- Project 'UI-MVC'
  - Properties:
    - naam v/d assembly 'SC.UI.Web.MVC'
    - standaard namespace 'SC.UI.Web.MVC'

- Wat met de reeds gecreëerde controllers, views, configuration…?
  ➔Find/Replace !!
  (=>'Rename' werkt hier niet!!)

Standaard namespace

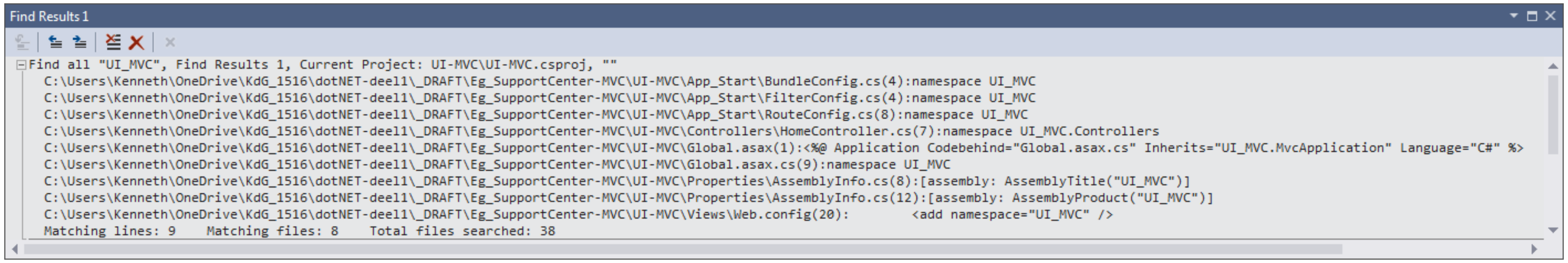
- Find/Replace


UI_MVC

SC.UI.Web.MVC

Current Project

➔Find All


Find Results 1

```
⊟Find all "UI_MVC", Find Results 1, Current Project: UI-MVC\UI-MVC.csproj, ""
  C:\Users\Kenneth\OneDrive\KdG_1516\dotNET-deel1\_DRAFT\Eg_SupportCenter-MVC\UI-MVC\App_Start\BundleConfig.cs(4):namespace UI_MVC
  C:\Users\Kenneth\OneDrive\KdG_1516\dotNET-deel1\_DRAFT\Eg_SupportCenter-MVC\UI-MVC\App_Start\FilterConfig.cs(4):namespace UI_MVC
  C:\Users\Kenneth\OneDrive\KdG_1516\dotNET-deel1\_DRAFT\Eg_SupportCenter-MVC\UI-MVC\App_Start\RouteConfig.cs(8):namespace UI_MVC
  C:\Users\Kenneth\OneDrive\KdG_1516\dotNET-deel1\_DRAFT\Eg_SupportCenter-MVC\UI-MVC\Controllers\HomeController.cs(7):namespace UI_MVC.Controllers
  C:\Users\Kenneth\OneDrive\KdG_1516\dotNET-deel1\_DRAFT\Eg_SupportCenter-MVC\UI-MVC\Global.asax(1):<%@ Application Codebehind="Global.asax.cs" Inherits="UI_MVC.MvcApplication" Language="C#" %>
  C:\Users\Kenneth\OneDrive\KdG_1516\dotNET-deel1\_DRAFT\Eg_SupportCenter-MVC\UI-MVC\Global.asax.cs(9):namespace UI_MVC
  C:\Users\Kenneth\OneDrive\KdG_1516\dotNET-deel1\_DRAFT\Eg_SupportCenter-MVC\UI-MVC\Properties\AssemblyInfo.cs(8):[assembly: AssemblyTitle("UI_MVC")]
  C:\Users\Kenneth\OneDrive\KdG_1516\dotNET-deel1\_DRAFT\Eg_SupportCenter-MVC\UI-MVC\Properties\AssemblyInfo.cs(12):[assembly: AssemblyProduct("UI_MVC")]
  C:\Users\Kenneth\OneDrive\KdG_1516\dotNET-deel1\_DRAFT\Eg_SupportCenter-MVC\UI-MVC\Views\Web.config(20):        <add namespace="UI_MVC" />
  Matching lines: 9    Matching files: 8    Total files searched: 38
```

➔(Replace All)

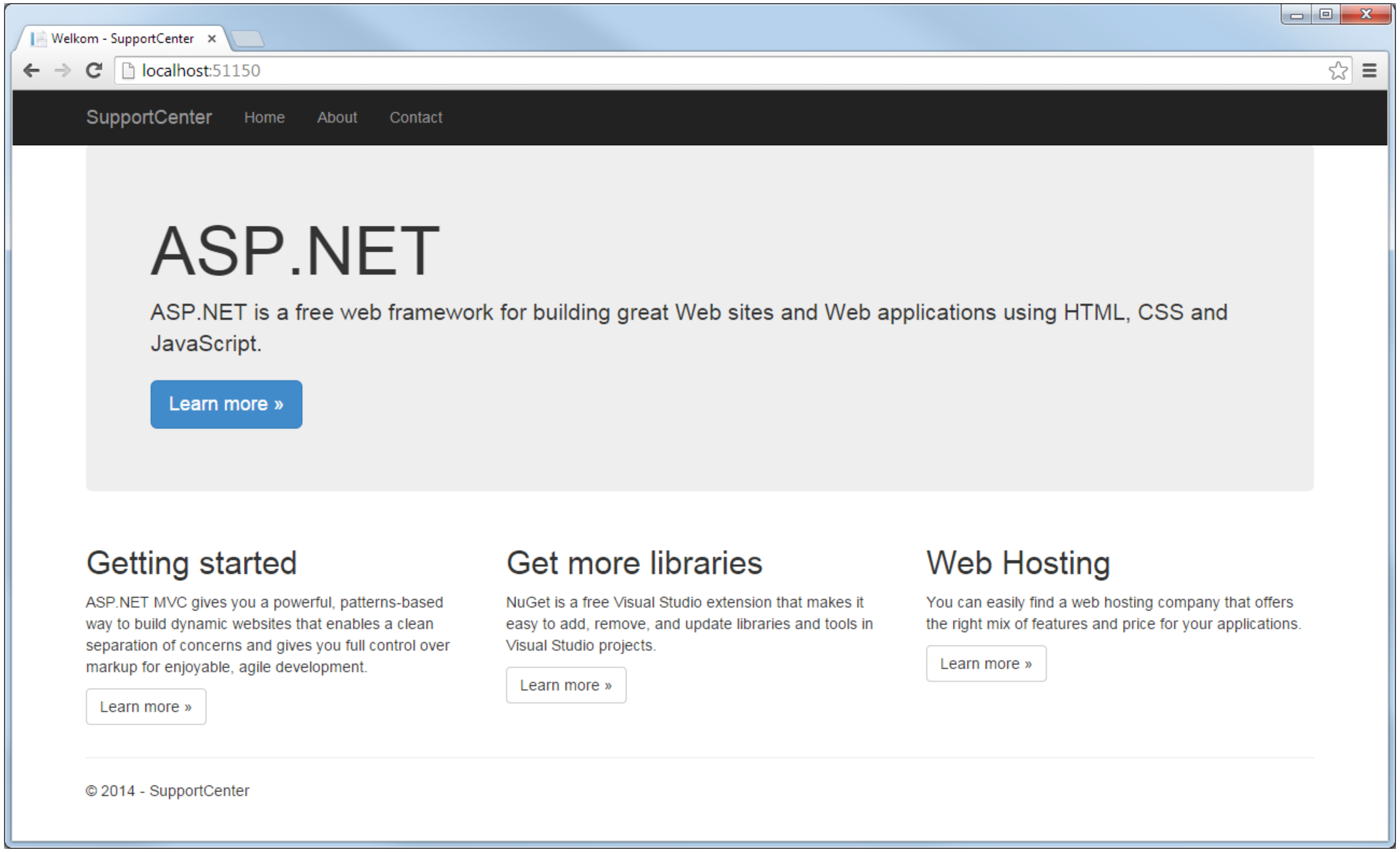
Connectionstring

- Voeg een connectionstring toe aan de **Web.config** (in de root van het project)
  - Naam: SupportCenterDB_EFCodeFirst
  - SQL Server: .\SQLSERVER2016
  - Databank: SupportCenterDB_EFCodeFirst

```
...
 <connectionStrings>
    ...
    <add name="SupportCenterDB-EFCodeFirst"
         connectionString="Data Source=.\SQLSERVER2016;
                           Initial Catalog=SupportCenterDB_EFCodeFirst;
                           Integrated Security=True"
         providerName="System.Data.SqlClient"/>
  </connectionStrings>
...
```

Oefening

- Run application (url?)



ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript.

Getting started

ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and gives you full control over markup for enjoyable, agile development.

Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

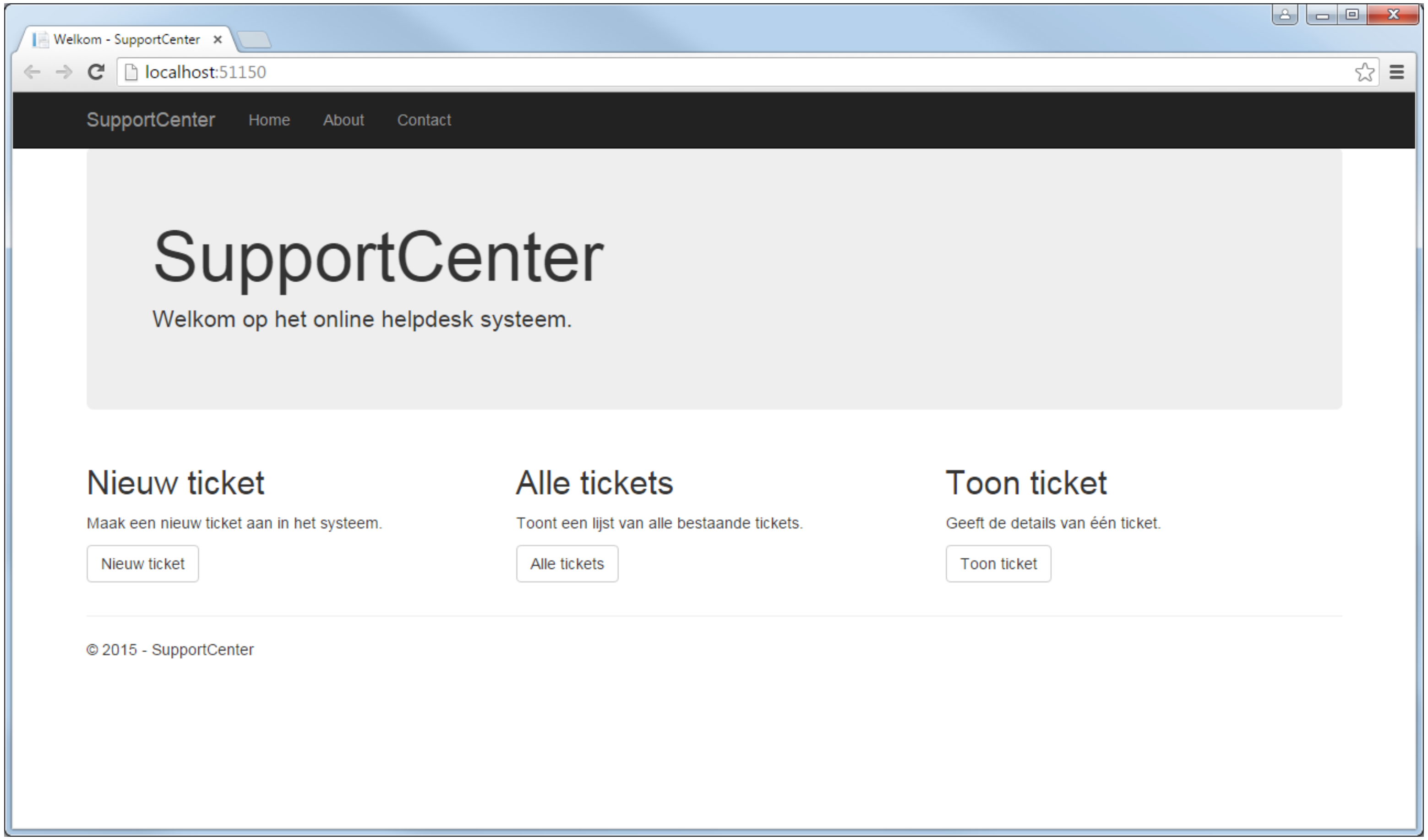
Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your applications.

© 2014 - SupportCenter

Oefening

- Wijzig de home-pagina naar:



Welkom - SupportCenter

localhost:51150

SupportCenter Home About Contact

SupportCenter

Welkom op het online helpdesk systeem.

Nieuw ticket

Maak een nieuw ticket aan in het systeem.

Nieuw ticket

Alle tickets

Toont een lijst van alle bestaande tickets.

Alle tickets

Toon ticket

Geeft de details van één ticket.

Toon ticket

© 2015 - SupportCenter

\Views\Shared_Layout.cshtml

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - SupportCenter</title>
    @Styles.Render("~/Content/css")
    @Scripts.Render("~/bundles/modernizr")
</head>
<body>
    <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>
                @Html.ActionLink("SupportCenter", "Index", "Home", new { area = "" }
                                                                      , new { @class = "navbar-brand" })
            </div>
```

\Views\Shared_Layout.cshtml

```
            <div class="navbar-collapse collapse">
                <ul class="nav navbar-nav">
                    <li>@Html.ActionLink("Home", "Index", "Home")</li>
                    <li>@Html.ActionLink("About", "About", "Home")</li>
                    <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
                </ul>
            </div>
        </div>
    </div>
    <div class="container body-content">
        @RenderBody()
        <hr />
        <footer>
            <p>&copy; @DateTime.Now.Year - SupportCenter</p>
        </footer>
    </div>

    @Scripts.Render("~/bundles/jquery")
    @Scripts.Render("~/bundles/bootstrap")
    @RenderSection("scripts", required: false)
</body>
</html>
```

\Views\Home\Index.cshtml

```
@{
    ViewBag.Title = "Welkom";
}

<div class="jumbotron">
    <h1>SupportCenter</h1>
    <p class="lead">Welkom op het online helpdesk systeem.</p>
</div>

<div class="row">
    <div class="col-md-4">
        <h2>Nieuw ticket</h2>
        <p>Maak een nieuw ticket aan in het systeem.</p>
        <p><a class="btn btn-default" href="#">Nieuw ticket</a></p>
    </div>
    <div class="col-md-4">
        <h2>Alle tickets</h2>
        <p>Toont een lijst van alle bestaande tickets.</p>
        <p><a class="btn btn-default" href="#">Alle tickets</a></p>
    </div>
    <div class="col-md-4">
        <h2>Toon ticket</h2>
        <p>Geeft de details van één ticket.</p>
        <p><a class="btn btn-default" href="#">Toon ticket</a></p>
    </div>
</div>
```

Oefening

- Voorzie een nieuwe controller 'TicketController', en maak gebruik van de template 'MVC5 Controller with read/write actions'
  - Voorzie een private veld 'mgr' van het type 'ITicketManager' en initialiseer met 'TicketManager'

TicketController.cs

```
...
using System.Web;
using System.Web.Mvc;

using SC.BL;
namespace SC.UI.Web.MVC.Controllers
{
  public class TicketController : Controller
  {
    private ITicketManager mgr = new TicketManager();

    // GET: Ticket
    public ActionResult Index()
    {
      return View();
    }

    // GET: Ticket/Details/5
    public ActionResult Details(int id)
    {
      return View();
    }
```

Karel de Grote Hogeschool

TicketController.cs

```
// GET: Ticket/Create
public ActionResult Create()
{
  return View();
}

// POST: Ticket/Create
[HttpPost]
public ActionResult Create(FormCollection collection)
{
  try
  {
    // TODO: Add insert logic here

    return RedirectToAction("Index");
  }
  catch
  {
    return View();
  }
}
```

TicketController.cs

```
// GET: Ticket/Edit/5
public ActionResult Edit(int id)
{
  return View();
}

// POST: Ticket/Edit/5
[HttpPost]
public ActionResult Edit(int id, FormCollection collection)
{
  try
  {
    // TODO: Add update logic here

    return RedirectToAction("Index");
  }
  catch
  {
    return View();
  }
}
```

TicketController.cs

```
    // GET: Ticket/Delete/5
    public ActionResult Delete(int id)
    {
      return View();
    }

    // POST: Ticket/Delete/5
    [HttpPost]
    public ActionResult Delete(int id, FormCollection collection)
    {
      try
      {
        // TODO: Add delete logic here

        return RedirectToAction("Index");
      }
      catch
      {
        return View();
      }
    }
  }
}
```

Oefening

- Werk de actionmethode 'Index' van de 'TicketController' uit
  - Vraag alle tickets op en geef door aan de view
  - Voorzie een view
    - Template: 'List'
    - Model class: 'SC.BL.Domain.Ticket'

- View:
  - Ticket.State?
    - type 'enum' wordt niet automatisch mee in view opgenomen -> zelf toevoegen!
  - Wat is het probleem met de hyperlinks 'Edit', 'Details' en 'Delete'?

TicketController.cs

```
...
using SC.BL;
using SC.BL.Domain;
namespace SC.UI.Web.MVC.Controllers
{
  public class TicketController : Controller
  {
    private TicketManager mgr = new TicketManager();

    // GET: Ticket
    public ActionResult Index()
    {
      IEnumerable<Ticket> tickets = mgr.GetTickets();
      return View(tickets);
    }

    ...
  }
}
```

\Views\Ticket\Index.cs

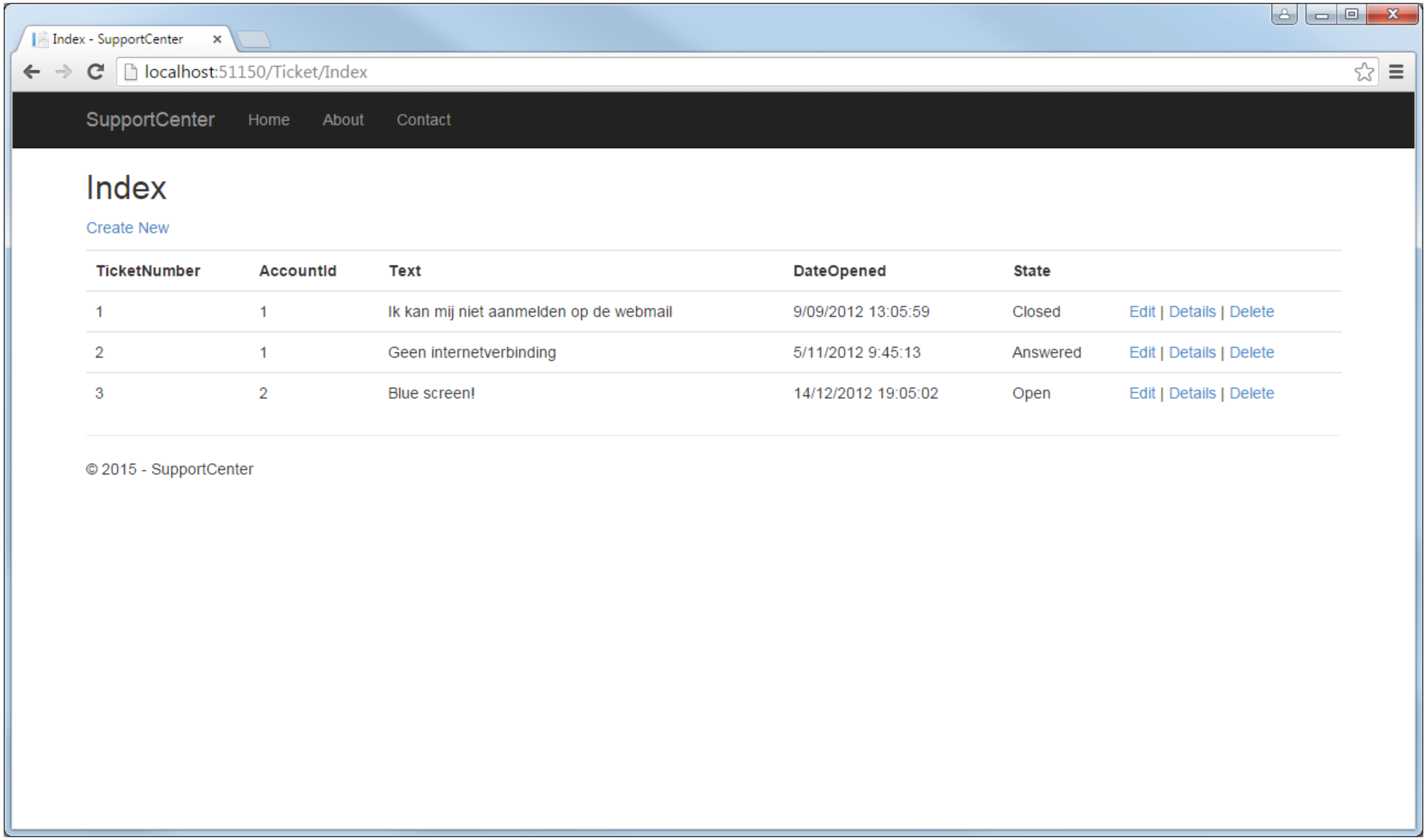
```
@model IEnumerable<SC.BL.Domain.Ticket>

@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>

<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.TicketNumber)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.AccountId)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Text)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.DateOpened)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.State)
        </th>
        <th></th>
    </tr>
```

\Views\Ticket\Index.cs

```
@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.TicketNumber)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.AccountId)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Text)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.DateOpened)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.State)
        </td>
        <td>
            @Html.ActionLink("Edit", "Edit", new { id=item.TicketNumber }) |
            @Html.ActionLink("Details", "Details", new { id=item.TicketNumber }) |
            @Html.ActionLink("Delete", "Delete", new { id=item.TicketNumber })
        </td>
    </tr>
}

</table>
```

Wijzig /* id=item.PrimaryKey */ naar id=item.TicketNumber
Oorzaak: vanuit het model 'Ticket' kan geen 'unique identifier' afgeleid worden
Alternatieve oplossing: voorzie het KeyAttribute op Ticket.TicketNumber

Resultaat



Index

Create New

TicketNumber	AccountId	Text	DateOpened	State	
1	1	Ik kan mij niet aanmelden op de webmail	9/09/2012 13:05:59	Closed	Edit | Details | Delete
2	1	Geen internetverbinding	5/11/2012 9:45:13	Answered	Edit | Details | Delete
3	2	Blue screen!	14/12/2012 19:05:02	Open	Edit | Details | Delete

© 2015 - SupportCenter

Oefening

- Werk de actionmethode 'Details' van de 'TicketController' uit
  - Vraag het ticket op voor 'id' en geef door aan de view
  - Voorzie een view
    - Template: 'Details'
    - Model class: 'SC.BL.Domain.Ticket'

- View:
  - Ticket.State
  - Hyperlink 'Edit'

TicketController.cs

```
...
using SC.BL;
using SC.BL.Domain;
namespace SC.UI.Web.MVC.Controllers
{
  public class TicketController : Controller
  {
    private TicketManager mgr = new TicketManager();

    ...

    // GET: Ticket/Details/5
    public ActionResult Details(int id)
    {
      Ticket ticket = mgr.GetTicket(id);
      return View(ticket);
    }

    ...
  }
}
```

Karel de Grote Hogeschool

\Views\Ticket\Details.cs

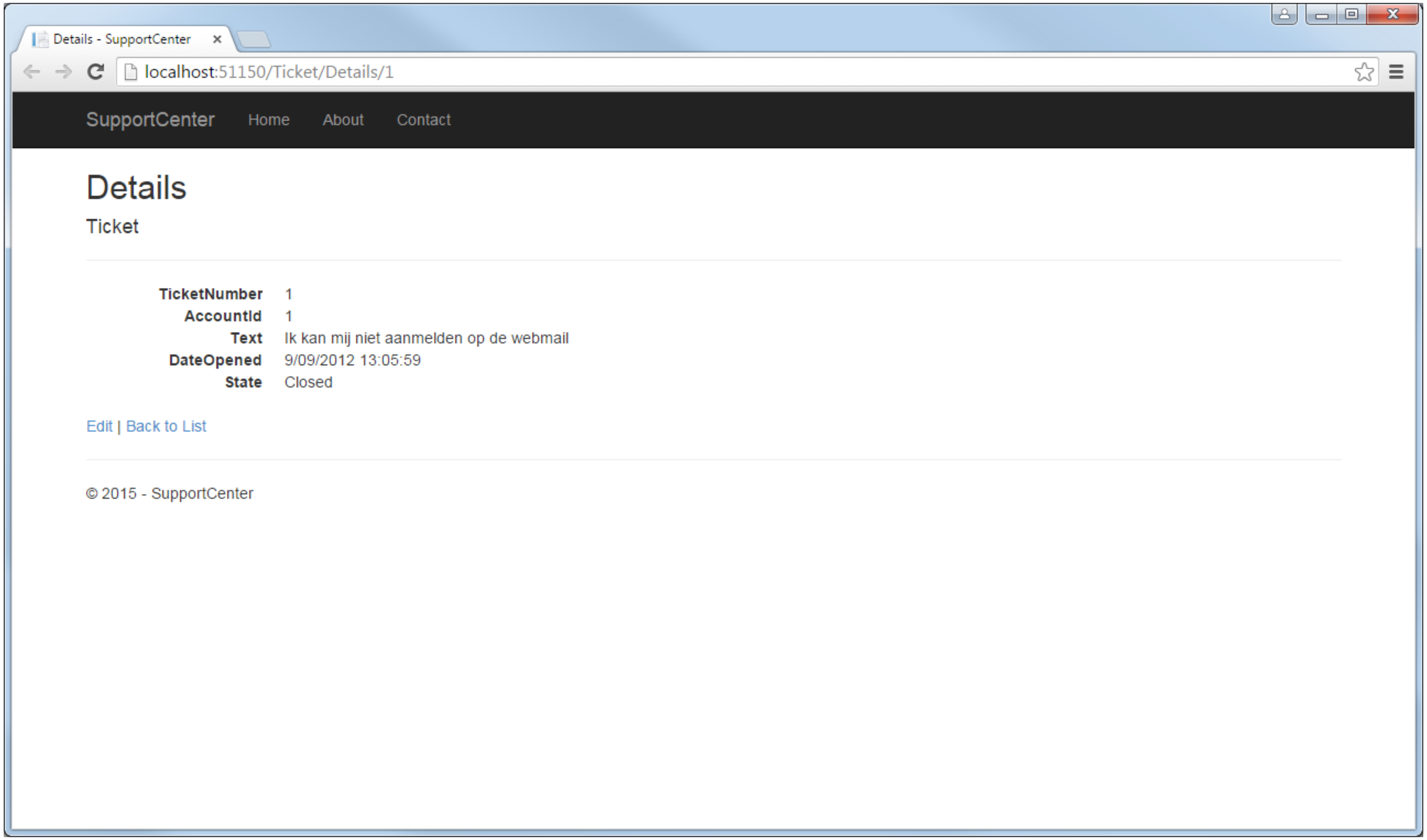
```
@model SC.BL.Domain.Ticket

@{
    ViewBag.Title = "Details";
}

<h2>Details</h2>

<div>
    <h4>Ticket</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.TicketNumber)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.TicketNumber)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.AccountId)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.AccountId)
        </dd>
```

\Views\Ticket\Details.cs

```
        <dt>
            @Html.DisplayNameFor(model => model.Text)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Text)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.DateOpened)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.DateOpened)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.State)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.State)
        </dd>

    </dl>
</div>
<p>
    @Html.ActionLink("Edit", "Edit", new { id = Model.TicketNumber }) |
    @Html.ActionLink("Back to List", "Index")
</p>
```

Karel de Grote Hogeschool

Resultaat



Details

Ticket

TicketNumber	1
AccountId	1
Text	Ik kan mij niet aanmelden op de webmail
DateOpened	9/09/2012 13:05:59
State	Closed

Edit | Back to List

© 2015 - SupportCenter

Oefening

- Werk de actionmethode 'Edit' van de 'TicketController' uit
  - HttpGet
    - Vraag het ticket op voor 'id' en geef door aan de view
  - HttpPost
    - Wijzig parameter 'collection' (FormCollection) naar 'ticket' (Ticket)
  - Voorzie een view
    - Template: 'Edit'
    - Model class: 'SC.BL.Domein.Ticket'

- View: 'TicketNumber' en 'State' !
  - TicketNumber mag niet wijzigbaar zijn, maar moet wel blijven bestaan, want moet mee met de http-post teruggestuurd worden ➔ Html.HiddenFor
  - State? ➔ Html.EnumDropDownListFor

TicketController.cs

```
...
  public class TicketController : Controller
  {
    private TicketManager mgr = new TicketManager();
    ...

    // GET: Ticket/Edit/5
    public ActionResult Edit(int id)
    {
      Ticket ticket = mgr.GetTicket(id);
      return View(ticket);
    }

    // POST: Ticket/Edit/5
    [HttpPost]
    public ActionResult Edit(int id, Ticket ticket)
    {
      try
      {
        mgr.ChangeTicket(ticket);

        return RedirectToAction("Index");
      }
      catch
      {
        return View();
      }
    }

    ...
  }
```

Karel de Grote Hogeschool

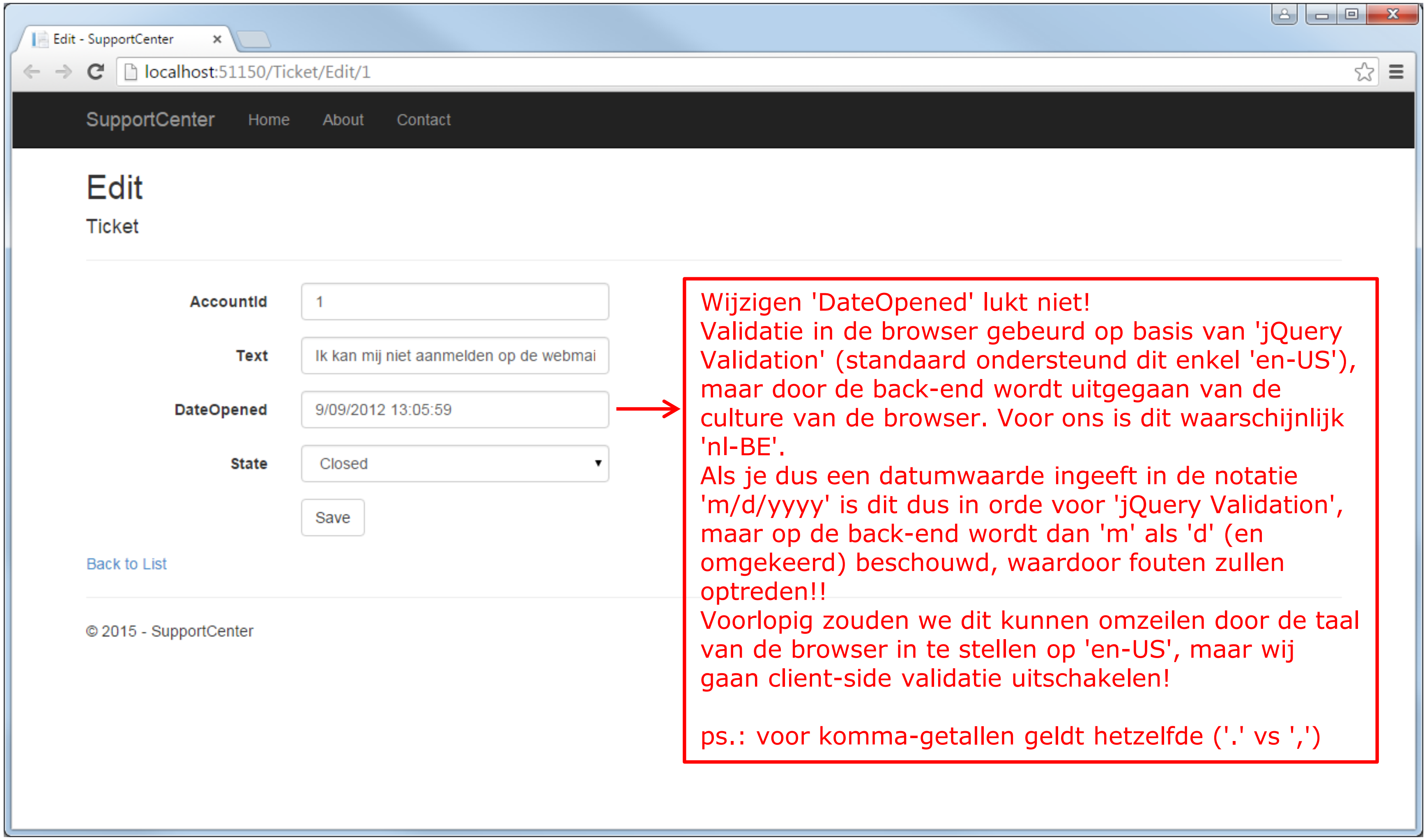
\Views\Ticket\Edit.cs

```
@model SC.BL.Domain.Ticket

@{
    ViewBag.Title = "Edit";
}

<h2>Edit</h2>


@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Ticket</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.TicketNumber)
        ~~<div class="form-group">~~
            ~~@Html.LabelFor(model => model.TicketNumber, htmlAttributes: new { @class = "control-label col-md-2" })~~
            ~~<div class="col-md-10">~~
                ~~@Html.EditorFor(model => model.TicketNumber, new { htmlAttributes = new { @class = "form-control" } })~~
                ~~@Html.ValidationMessageFor(model => model.TicketNumber, "", new { @class = "text-danger" })~~
            ~~</div>~~
        ~~</div>~~

        <div class="form-group">
            @Html.LabelFor(model => model.AccountId, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.AccountId, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.AccountId, "", new { @class = "text-danger" })
            </div>
        </div>
```

\Views\Ticket\Edit.cs

```
        <div class="form-group">
            @Html.LabelFor(model => model.Text, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Text, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Text, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.DateOpened, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.DateOpened, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.DateOpened, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.State, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EnumDropDownListFor(model => model.State, htmlAttributes: new { @class = "form-control" })
                @Html.ValidationMessageFor(model => model.State, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Save" class="btn btn-default" />
            </div>
        </div>
    </div>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

Resultaat


Edit - SupportCenter

localhost:51150/Ticket/Edit/1

Edit

Ticket

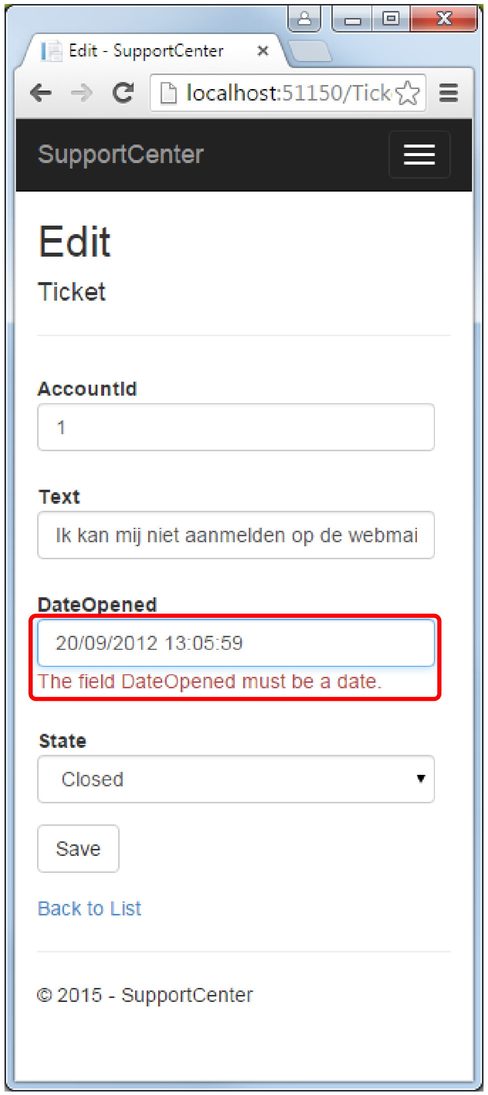
AccountId	1
Text	Ik kan mij niet aanmelden op de webmai
DateOpened	9/09/2012 13:05:59
State	Closed

© 2015 - SupportCenter

Wijzigen 'DateOpened' lukt niet!
Validatie in de browser gebeurd op basis van 'jQuery Validation' (standaard ondersteund dit enkel 'en-US'), maar door de back-end wordt uitgegaan van de culture van de browser. Voor ons is dit waarschijnlijk 'nl-BE'.
Als je dus een datumwaarde ingeeft in de notatie 'm/d/yyyy' is dit dus in orde voor 'jQuery Validation', maar op de back-end wordt dan 'm' als 'd' (en omgekeerd) beschouwd, waardoor fouten zullen optreden!!
Voorlopig zouden we dit kunnen omzeilen door de taal van de browser in te stellen op 'en-US', maar wij gaan client-side validatie uitschakelen!

ps.: voor komma-getallen geldt hetzelfde ('.' vs ',')

Resultaat – validatie


Edit - SupportCenter

localhost:51150/Tick

SupportCenter

Edit

Ticket

AccountId

1

Text

Ik kan mij niet aanmelden op de webmai

DateOpened

20/09/2012 13:05:59

The field DateOpened must be a date.

State

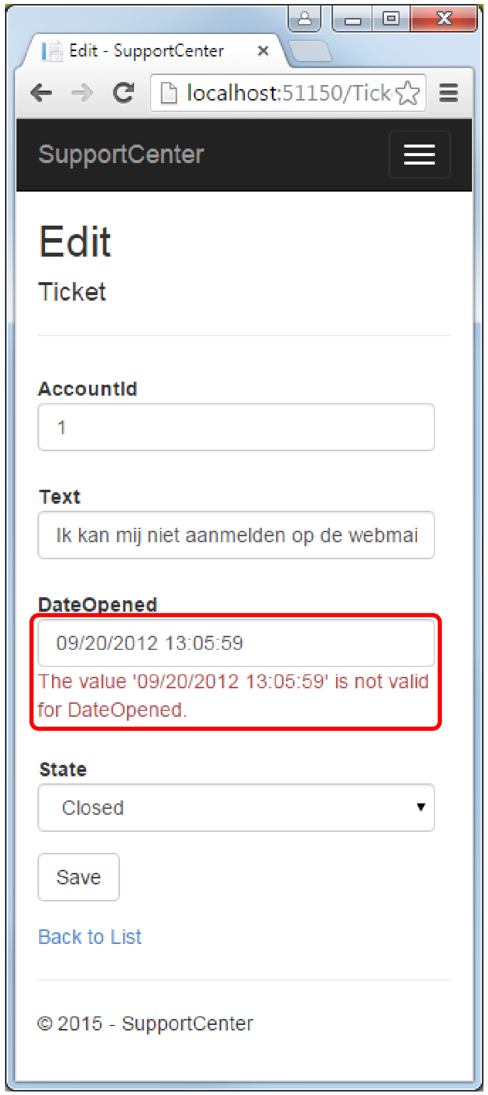
Closed

Save

Back to List

© 2015 - SupportCenter

client-side validatie
(adhv JQuery Validation)


Edit - SupportCenter

localhost:51150/Tick

SupportCenter

Edit

Ticket

AccountId

1

Text

Ik kan mij niet aanmelden op de webmai

DateOpened

09/20/2012 13:05:59

The value '09/20/2012 13:05:59' is not valid for DateOpened.

State

Closed

Save

Back to List

© 2015 - SupportCenter

back-end validatie
(adhv Culture browser)

Client-side validatie uitschakelen

- Web.config

```
<configuration>
  ...
  <appSettings>
    ...
    <add key="ClientValidationEnabled" value="false" />
    ...
  </appSettings>
  ...
</configuration>
```

Oefening

- Werk de actionmethode 'Delete' van de 'TicketController' uit
  - HttpGet
    - Vraag het ticket op voor 'id' en geef door aan de view
  - HttpPost
    - Parameter 'collection' (FormCollection) wordt niet gebruikt, maar we kiezen er voor om deze niet te verwijderen, waarom?
      - get- en post-actionmethode hebben dan dezelfde signatuur (overload), wat problemen geeft voor het routen van de url naar de actionmethode door MVC
  - Voorzie een view
    - Template: 'Delete'
    - Model class: 'SC.BL.Domein.Ticket'

- View: State!

TicketController.cs

```
...
  public class TicketController : Controller
  {
    private TicketManager mgr = new TicketManager();
    ...

    // GET: Ticket/Delete/5
    public ActionResult Delete(int id)
    {
      Ticket ticket = mgr.GetTicket(id);
      return View(ticket);
    }

    // POST: Ticket/Delete/5
    [HttpPost]
    public ActionResult Delete(int id, ~~FormCollection collection~~)
    {
      try
      {
        mgr.RemoveTicket(id);

        return RedirectToAction("Index");
      }
      catch
      {
        return View();
      }
    }

    ...
  }
```

\Views\Ticket\Delete.cs

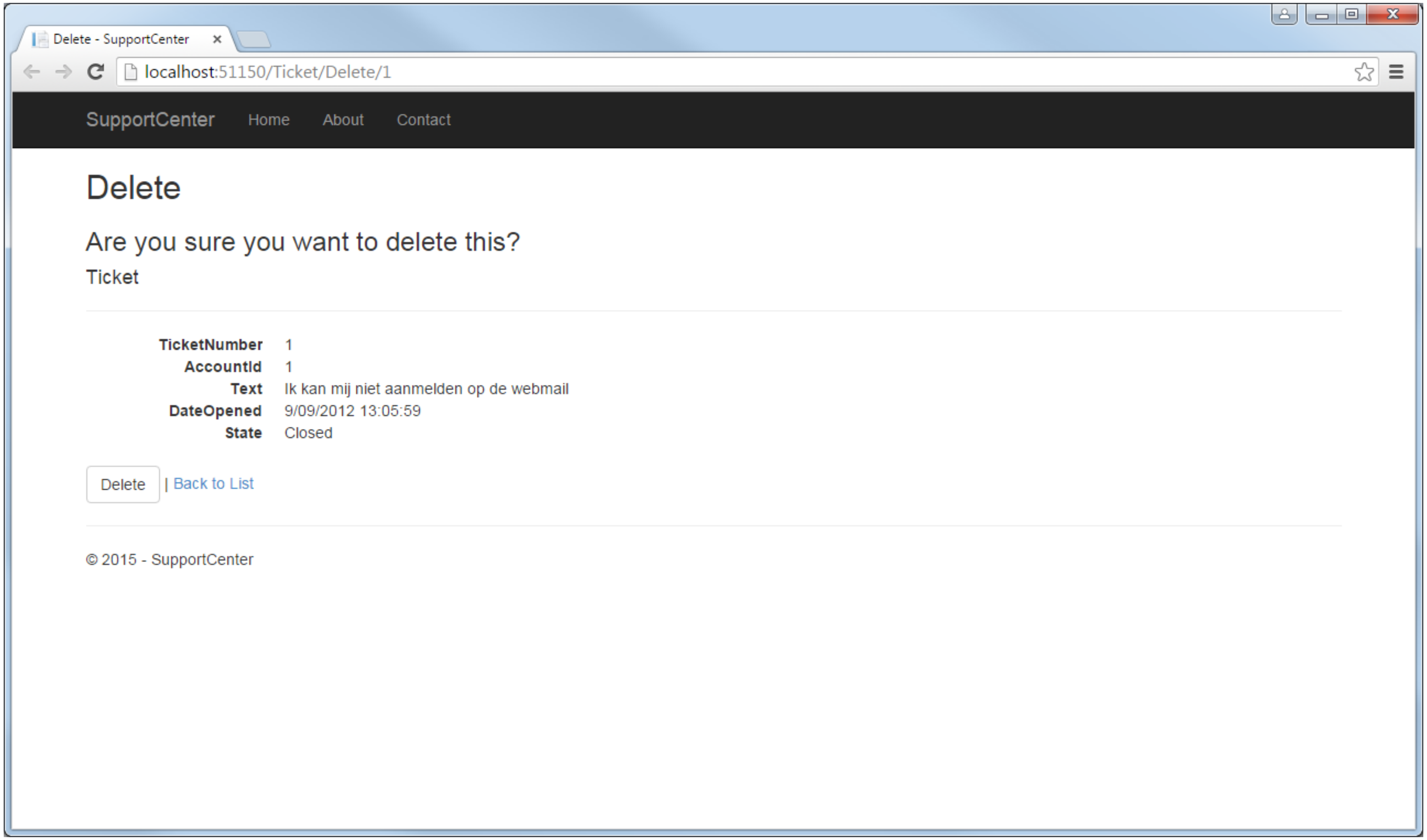
```
@model SC.BL.Domain.Ticket
@{
    ViewBag.Title = "Delete";
}
<h2>Delete</h2>
<h3>Are you sure you want to delete this?</h3>
<div>
    <h4>Ticket</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.TicketNumber)
        </dt>
        <dd>
            @Html.DisplayFor(model => model.TicketNumber)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.AccountId)
        </dt>
        <dd>
            @Html.DisplayFor(model => model.AccountId)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.Text)
        </dt>
        <dd>
            @Html.DisplayFor(model => model.Text)
        </dd>
```

\Views\Ticket\Delete.cs

```
        <dt>
            @Html.DisplayNameFor(model => model.DateOpened)
        </dt>
        <dd>
            @Html.DisplayFor(model => model.DateOpened)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.State)
        </dt>
        <dd>
            @Html.DisplayFor(model => model.State)
        </dd>

    </dl>
    @using (Html.BeginForm()) {
        @Html.AntiForgeryToken()

        <div class="form-actions no-color">
            <input type="submit" value="Delete" class="btn btn-default" /> |
            @Html.ActionLink("Back to List", "Index")
        </div>
    }
</div>
```

Karel de Grote Hogeschool

Resultaat



Delete

Are you sure you want to delete this?

Ticket

TicketNumber	1
AccountId	1
Text	Ik kan mij niet aanmelden op de webmail
DateOpened	9/09/2012 13:05:59
State	Closed

© 2015 - SupportCenter

Oefening

- Werk de actionmethode 'Create' van de 'TicketController' uit
  - HttpGet
  - HttpPost
    - Wijzig parameter 'collection' (FormCollection) naar 'ticket' (Ticket)
    - Redirect naar details-pagina van het nieuwe ticket
  - Voorzie een view
    - Template: 'Create'
    - Model class: 'SC.BL.Domein.Ticket'

- View:
  - Verwijder 'TicketNumber' en 'DateOpened'

TicketController.cs

```
...
 public class TicketController : Controller
 {
   private TicketManager mgr = new TicketManager();
   ...

   // GET: Ticket/Create
   public ActionResult Create()
   {
     return View();
   }

   // POST: Ticket/Create
   [HttpPost]
   public ActionResult Create(Ticket ticket)
   {
     try
     {
       ticket = mgr.AddTicket(ticket.AccountId, ticket.Text);

       return RedirectToAction("Details", new { id = ticket.TicketNumber });
     }
     catch
     {
       return View();
     }
   }

   ...
 }
```

Karel de Grote Hogeschool

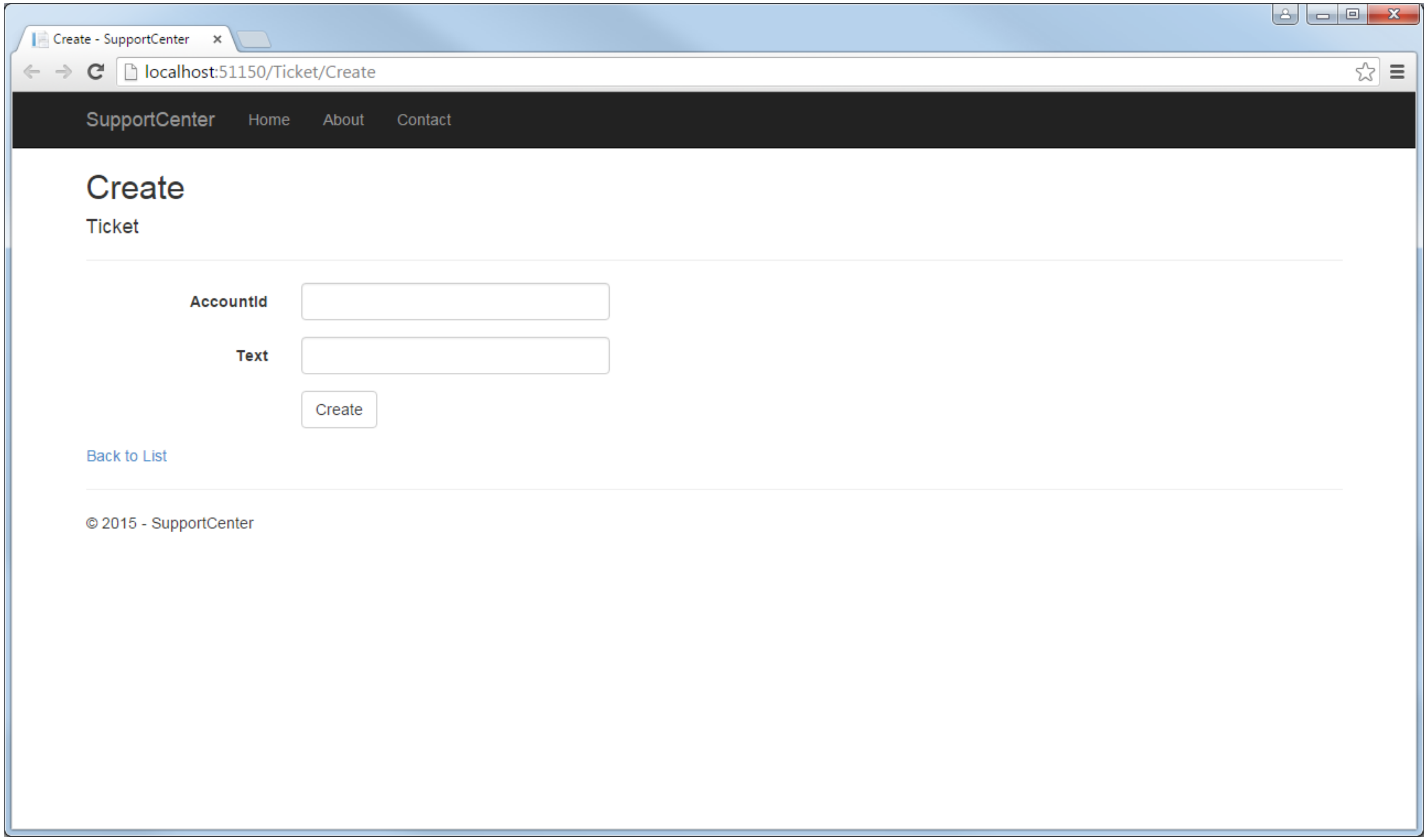
\Views\Ticket\Create.cs

```
@model SC.BL.Domain.Ticket

@{
    ViewBag.Title = "Create";
}

<h2>Create</h2>


@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Ticket</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        ~~<div class="form-group">~~
            ~~@Html.LabelFor(model => model.TicketNumber, htmlAttributes: new { @class = "control-label col-md-2" })~~
            ~~<div class="col-md-10">~~
                ~~@Html.EditorFor(model => model.TicketNumber, new { htmlAttributes: new { @class = "form-control" } })~~
                ~~@Html.ValidationMessageFor(model => model.TicketNumber, "", new { @class = "text-danger" })~~
            ~~</div>~~
        ~~</div>~~

        <div class="form-group">
            @Html.LabelFor(model => model.AccountId, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.AccountId, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.AccountId, "", new { @class = "text-danger" })
            </div>
        </div>
```

\Views\Ticket\Create.cs

```
        <div class="form-group">
            @Html.LabelFor(model => model.Text, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Text, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Text, "", new { @class = "text-danger" })
            </div>
        </div>

        ~~<div class="form-group">~~
            ~~@Html.LabelFor(model => model.DateOpened, htmlAttributes: new { @class = "control-label col-md-2" })~~
            ~~<div class="col-md-10">~~
                ~~@Html.EditorFor(model => model.DateOpened, new { htmlAttributes = new { @class = "form-control" } })~~
                ~~@Html.ValidationMessageFor(model => model.DateOpened, "", new { @class = "text-danger" })~~
            ~~</div>~~
        ~~</div>~~

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Save" class="btn btn-default" />
            </div>
        </div>
    </div>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

Resultaat



Create - SupportCenter

localhost:51150/Ticket/Create

SupportCenter Home About Contact

Create

Ticket

AccountId

Text

Create

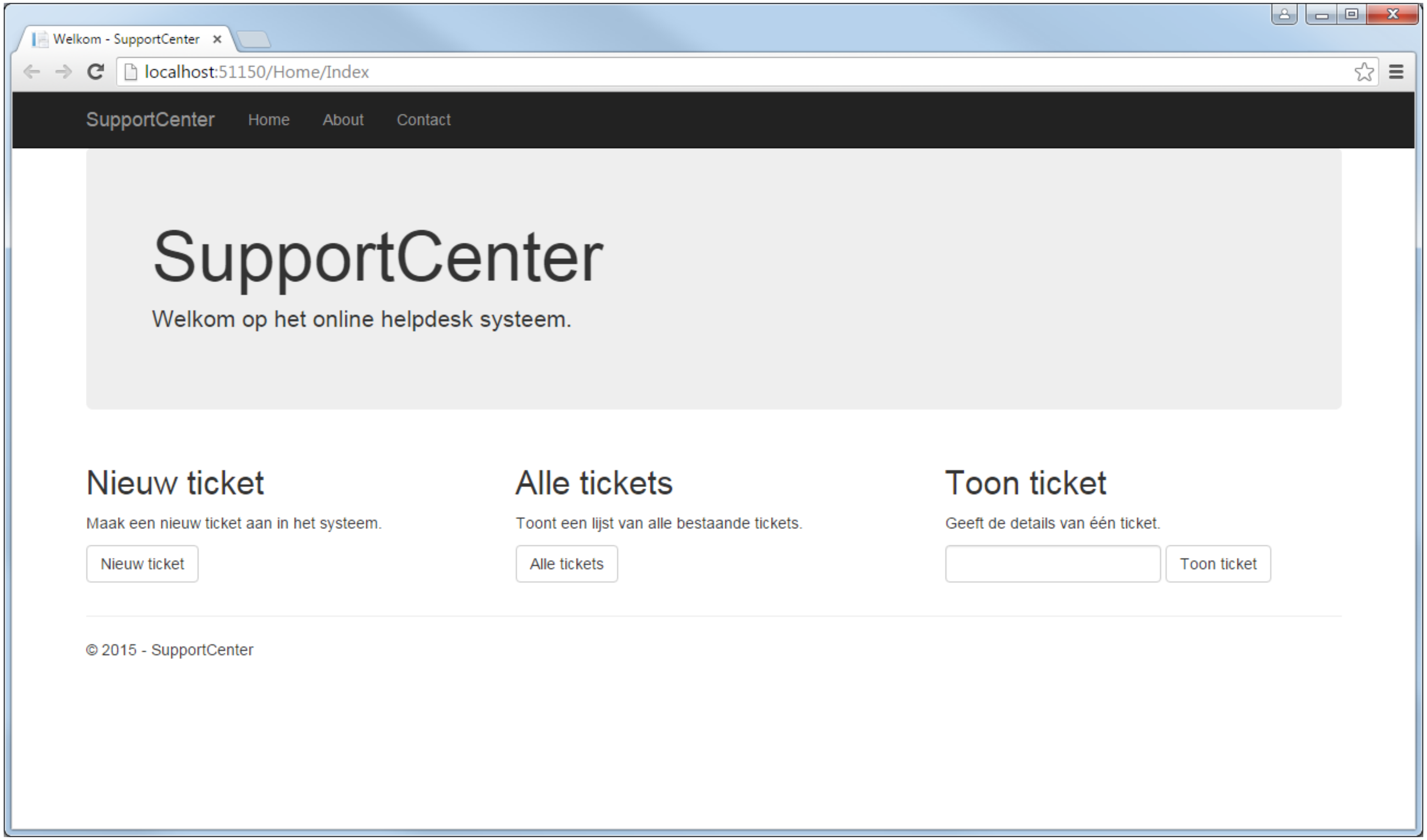
Back to List

© 2015 - SupportCenter

Oefening

- Zorg dat de knoppen op de home-pagina naar de juiste pagina navigeren
  - 'Nieuw ticket'
    - maak gebruik van Url.Action
  - 'Alle tickets'
    - maak gebruik van Html.ActionLink
  - 'Toon ticket'
    - voorzie een bijkomend tekst-veld om het nummer van het gewenste ticket in te kunnen geven (zie voorbeeld)
    - maak gebruik van klassieke html (form-, label-, en input-elementen) en css (bootstrap)

'Toon ticket': met form



Welkom - SupportCenter

localhost:51150/Home/Index

SupportCenter Home About Contact

SupportCenter

Welkom op het online helpdesk systeem.

Nieuw ticket

Maak een nieuw ticket aan in het systeem.

[Nieuw ticket]

Alle tickets

Toont een lijst van alle bestaande tickets.

[Alle tickets]

Toon ticket

Geeft de details van één ticket.

[Toon ticket]

© 2015 - SupportCenter

\Home\Index.cshtml

```
@{
    ViewBag.Title = "Welkom";
}

<div class="jumbotron">
    <h1>SupportCenter</h1>
    <p class="lead">Welkom op het online helpdesk systeem.</p>
</div>

<div class="row">
    <div class="col-md-4">
        <h2>Nieuw ticket</h2>
        <p>Maak een nieuw ticket aan in het systeem.</p>
        <p><a class="btn btn-default" href="@Url.Action("Create", "Ticket")">Nieuw ticket</a></p>
    </div>
    <div class="col-md-4">
        <h2>Alle tickets</h2>
        <p>Toont een lijst van alle bestaande tickets.</p>
        @Html.ActionLink("Alle tickets", "Index", "Ticket", null, new { @class = "btn btn-default" })
    </div>
    <div class="col-md-4">
        <h2>Toon ticket</h2>
        <p>Geeft de details van één ticket.</p>
        <form action="/Ticket/Details" method="get" class="form-inline">
            <div class="form-group">
                <label class="sr-only" form="id" />Ticketnummer</label>
                <input id="id" name="id" type="text" class="form-control">
            </div>
            <input type="submit" value="Toon ticket" class="btn btn-default">
        </form>
    </div>
</div>
```

Oefening

- Wijzig de uitwerking van het gedeelte 'Toon ticket' naar een uitwerking m.b.v. html-helpers
  - Html.BeginForm, Html.Label en Html.TextBox

\Home\Index.cshtml

```
@{
    ViewBag.Title = "Welkom";
}

<div class="jumbotron">
    <h1>SupportCenter</h1>
    <p class="lead">Welkom op het online helpdesk systeem.</p>
</div>

<div class="row">
    <div class="col-md-4">
        <h2>Nieuw ticket</h2>
        <p>Maak een nieuw ticket aan in het systeem.</p>
        <p><a class="btn btn-default" href="@Url.Action("Create", "Ticket")">Nieuw ticket</a></p>
    </div>
    <div class="col-md-4">
        <h2>Alle tickets</h2>
        <p>Toont een lijst van alle bestaande tickets.</p>
        @Html.ActionLink("Alle tickets", "Index", "Ticket", null, new { @class = "btn btn-default" })
    </div>
    <div class="col-md-4">
        <h2>Toon ticket</h2>
        <p>Geeft de details van één ticket.</p>
        @using (Html.BeginForm("Details", "Ticket", FormMethod.Get, new { @class = "form-inline" })) {
            <div class="form-group">
                @Html.Label("id", "Ticketnummer", new { @class = "sr-only" })
                @Html.TextBox("id", null, new { @class = "form-control" })
            </div>
            <input type="submit" value="Toon ticket" class="btn btn-default" />
        }
    </div>
</div>
```

Oefening

- Breid de Details-pagina van een ticket uit met alle antwoorden op het ticket
  - Details-actionmethode: vraag ook de 'TicketResponses' van het ticket op
    - Optie: maak gebruik van de ViewBag
  - Voorzie in de map '\Views\Ticket' een partialview '_TicketResponsesPartial'
    - Template: 'List'
    - Model class: 'SC.BL.Domein.TicketResponse'
    - Verwijder alle hyperlinks
    - table-element: id 'responses'
  - Details-view
    - Voeg onderaan een titel 'Responses' (h4) toe, en toon daaronder de partialview '_TicketResponsesPartial'

TicketController.cs

```
...
using SC.BL;
using SC.BL.Domain;
namespace SC.UI.Web.MVC.Controllers
{
  public class TicketController : Controller
  {
    private TicketManager mgr = new TicketManager();

    ...

    // GET: Ticket/Details/5
    public ActionResult Details(int id)
    {
      Ticket ticket = mgr.GetTicket(id);

      ticket.Responses = new List<TicketResponse>(mgr.GetTicketResponses(id));
      // OF: via ViewBag
      ViewBag.Responses = new List<TicketResponse>(mgr.GetTicketResponses(id));

      return View(ticket);
    }

    ...
  }
}
```

KdG Karel de Grote Hogeschool

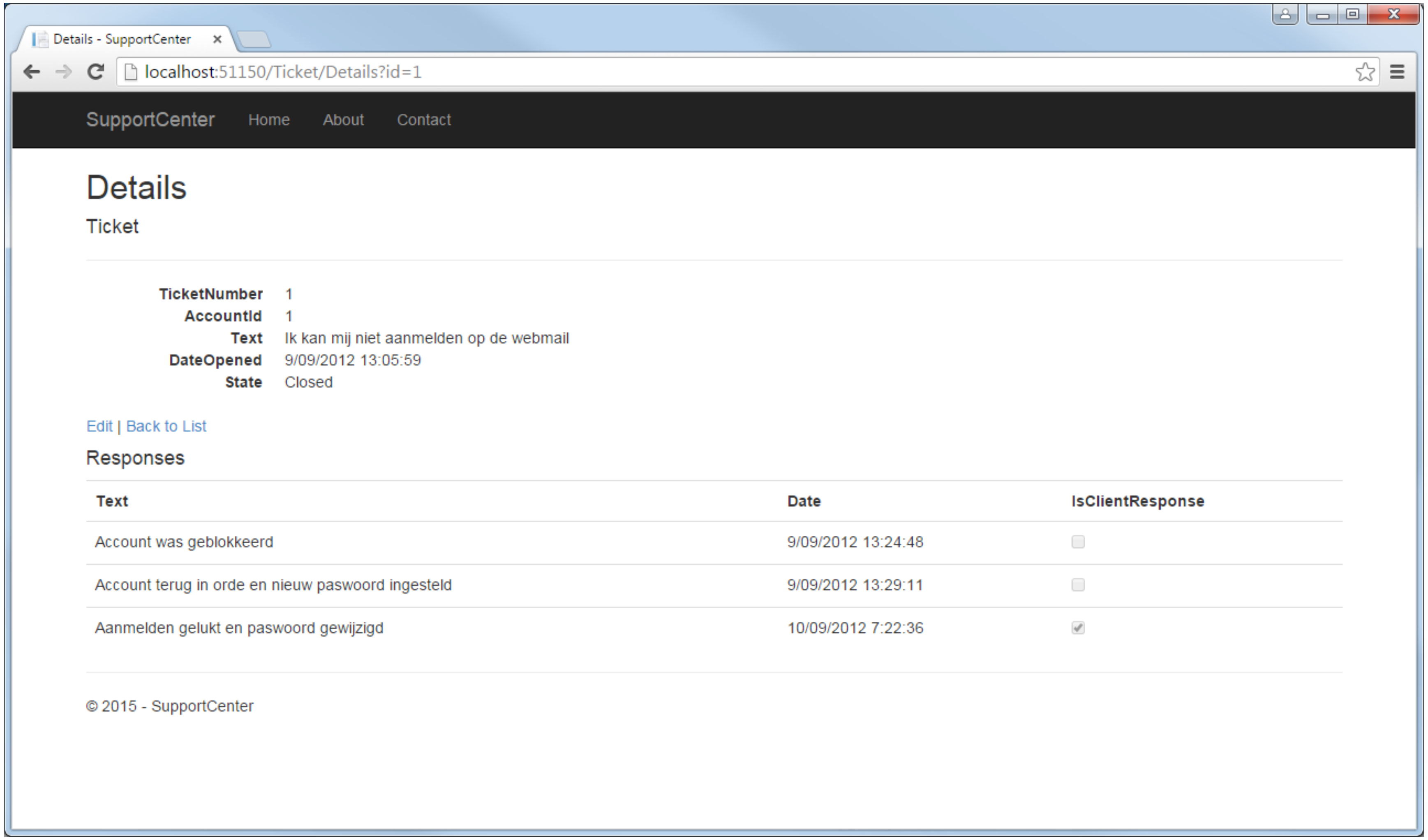
\Views\Ticket_TicketResponsesPartial.cs

```
@model IEnumerable<SC.BL.Domain.TicketResponse>

~~<p>~~
    ~~@Html.ActionLink("Create New", "Create")~~
~~</p>~~
<table id="responses" class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.Text)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Date)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.IsClientResponse)
        </th>
        ~~<th></th>~~
    </tr>

@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.Text)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Date)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.IsClientResponse)
        </td>
        ~~<td>~~
            ~~@Html.ActionLink("Edit", "Edit", new { id=item.Id }) |~~
            ~~@Html.ActionLink("Details", "Details", new { id=item.Id }) |~~
            ~~@Html.ActionLink("Delete", "Delete", new { id=item.Id })~~
        ~~</td>~~
    </tr>
}

</table>
```

\Views\Ticket\Details.cs

```
@model SC.BL.Domain.Ticket

@{
    ViewBag.Title = "Details";
}

<h2>Details</h2>

<div>
    <h4>Ticket</h4>
    <hr />
    <dl class="dl-horizontal">

        ...

    </dl>
</div>
<p>
    @Html.ActionLink("Edit", "Edit", new { id = Model.TicketNumber }) |
    @Html.ActionLink("Back to List", "Index")
</p>
<h4>Responses</h4>
@Html.Partial("_TicketResponsesPartial", Model.Responses)
@* /* OF: via ViewBag */ *@
@Html.Partial("_TicketResponsesPartial", (IEnumerable<SC.BL.Domain.TicketResponse>)ViewBag.Responses)
```

Resultaat



Details

Ticket

TicketNumber	1
AccountId	1
Text	Ik kan mij niet aanmelden op de webmail
DateOpened	9/09/2012 13:05:59
State	Closed

Edit | Back to List

Responses

Text	**Date**	**IsClientResponse**
Account was geblokkeerd	9/09/2012 13:24:48	☐
Account terug in orde en nieuw paswoord ingesteld	9/09/2012 13:29:11	☐
Aanmelden gelukt en paswoord gewijzigd	10/09/2012 7:22:36	✓

© 2015 - SupportCenter

Karel de Grote Hogeschool

Server-side validatie

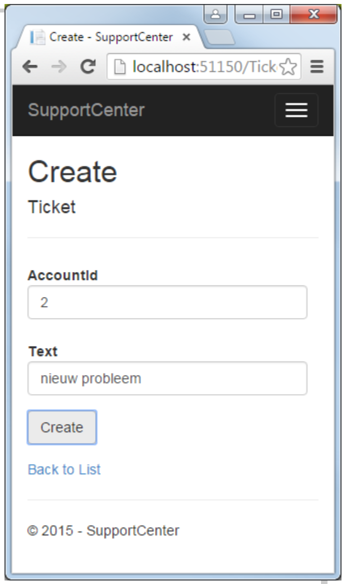
- Voorzie dat telkens de gebruiker zijn invoer 'submit', deze data op de back-end gevalideerd wordt
  - Het feit dat deze validatie ook wordt uitgevoerd door de TicketManager-klasse in de BL-laag negeren we hier even!
  - Bekijk de binnenkomende data en of deze als validate wordt beschouwd! => Create?

TicketController.cs – Edit

```
...
  public class TicketController : Controller
  {
    private TicketManager mgr = new TicketManager();

    ...

    // GET: Ticket/Edit/5
    public ActionResult Edit(int id)
    {
      Ticket ticket = mgr.GetTicket(id);
      return View(ticket);
    }

    // POST: Ticket/Edit/5
    [HttpPost]
    public ActionResult Edit(int id, Ticket ticket)
    {
      if (ModelState.isValid)
      {
        mgr.ChangeTicket(ticket);

        return RedirectToAction("Index");
      }

      return View();
    }

    ...
  }
```

TicketController.cs – Create

```
...
  public class TicketController : Controller
  {
    private TicketManager mgr = new TicketManager();

    ...

    // GET: Ticket/Create
    public ActionResult Create()
    {
      return View();
    }

    // POST: Ticket/Create
    [HttpPost]
    public ActionResult Create(Ticket ticket)
    {
      if (ModelState.IsValid)
      {
        ticket = mgr.AddTicket(ticket.AccountId, ticket.Text);

        return RedirectToAction("Details", new { id = ticket.TicketNumber });
      }

      return View();
    }

    ...
  }
```


Create - SupportCenter
localhost:51150/Tick
SupportCenter

Create

Ticket

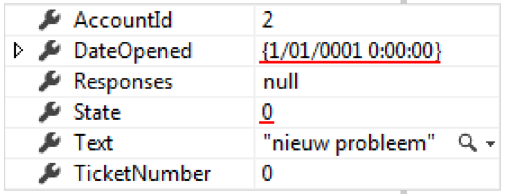
AccountId
2

Text
nieuw probleem

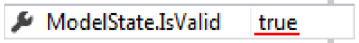
Create

Back to List

© 2015 - SupportCenter



AccountId	2
DateOpened	{1/01/0001 0:00:00}
Responses	null
State	0
Text	"nieuw probleem"
TicketNumber	0



ModelState.IsValid	true

KdG Karel de Grote Hogeschool

Oefening

- Bij het aanmaken van een nieuw ticket hebben we vastgesteld dat vanuit gebruikers-input kant enkel 'accountid' en 'text' noodzakelijke data is.
  In de controller is het datatype van de binnenkomende parameter echter 'Ticket'!
- We hebben ook vastgesteld dat de properties van Ticket waarvoor geen waarde met de http-request zijn meegegeven, worden ingesteld op de default-waarde van het betreffende datatype (via default constructor).
- Voorzie een specifiek *view*-model met de naam 'CreateTicketVM' (in de map 'Models') en gebruik dit om de Create-request te verwerken.
  Maak in dit model gebruik van volgende properties:
  - AccId (int)
  - Problem (string)

\Models\CreateTicketVM.cs

```
...

namespace SC.UI.Web.MVC.Models
{
  public class CreateTicketVM
  {
    public int AccId { get; set; }
    public int Problem { get; set; }
  }
}
```

Karel de Grote Hogeschool

TicketController.cs – Create

```
...
using SC.UI.Web.MVC.Models;
...
  public class TicketController : Controller
  {
    private TicketManager mgr = new TicketManager();

    ...

    // GET: Ticket/Create
    public ActionResult Create()
    {
      return View();
    }

    // POST: Ticket/Create
    [HttpPost]
    public ActionResult Create(CreateTicketVM newTicket)
    {
      if (ModelState.IsValid)
      {
        Ticket ticket = mgr.AddTicket(newTicket.AccountId, newTicket.Problem);

        return RedirectToAction("Details", new { id = ticket.TicketNumber });
      }

      return View();
    }

    ...
  }
```

\Views\Ticket\Create.cs

```
@model SC.UI.Web.MVC.Models.CreateTicketVM

@{
    ViewBag.Title = "Create";
}

<h2>Create</h2>


@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Ticket</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })

        <div class="form-group">
            @Html.LabelFor(model => model.AccId, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.AccId, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.AccId, "", new { @class = "text-danger" })
            </div>
        </div>
```

\Views\Ticket\Create.cs

```
        <div class="form-group">
            @Html.LabelFor(model => model.Problem, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Problem, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Problem, "", new { @class = "text-danger" })
            </div>
        </div>


        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Save" class="btn btn-default" />
            </div>
        </div>
    </div>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```