# Супермаркет – курсов проект по „Уеб приложения с ASP.NET MVC” – вариант #6

Да се проектира и реализира информационна система за управление на супермаркет за продажба на хранителни стоки. Системата трябва да поддържа продукти, разделени по категории и да следи наличностите и продажбите.

В супермаркета ежедневно се зарежда стока и постоянно се продават от наличните продукти.

## Потребители

1. Потребителите в системата играят ролята на касиери и могат да продават стока и да зареждат супермаркета.
2. Достъпът до системата става само след автентикация (login).
3. Специален потребител, наречен администратор, с username "**admin**" е единственият, който може да регистрира нови потребители, при което той въвежда следните данни за тях:

* username (съдържа само букви и цифри, уникално за системата)
* парола
* email
* име
* фамилия

## Категории

1. Една категория може да съдържа много продукти. Категориите имат единствено наимено­вание (например безалкохолни напитки, колбаси, хлебни изделия, захарни изделия и др.).
2. Потребителите могат да извършват следните действия с категории:

* добавяне на категория
* изтриване на категория – позволява се само ако няма нито един продукт в нея
* преименуване на категория

## Продукти

1. За всеки продукт се поддържа следната информация:

* наименование (например "бира Загорка")
* производител (например "Загорка" АД)
* мерна единица (например литър, килограм, брой, ...)
* категория (например безалкохолни напитки)
* цена (продажна цена към крайния клиент заедно с търговката надценка)

1. Потребителите (касиерите) могат да извършват следните действия с продуктите:

* добавяне на продукт
* изтриване на продукт – при изтриване продуктът се деактивира, за да може да участва в справките, но не е достъпен при останалите операции
* редактиране на продукт

1. Потребителите (касиерите) могат да променят цените на всеки продукт, независимо какви наличности има от него в момента и на какви цени са постъпили от доставчика.

## Каса

1. За супермаркета се поддържа каса с пари. При продажба парите в касата се увеличават, а наличностите на купените продукти намаляват. При зареждане на стока парите в касата намаляват и наличностите се увеличават.
2. Наличността в касата трябва да се показва на всяка страница след като потребителят се е автентикирал.
3. В начално, при първоначална инсталация на системата, в касата трябва да има 1 000 (хиляда) лева, с които да може да се зареди стока и след това да се започнат продажбите.

## Продажба на стока

1. Всяка продажба, реализирана в супермаркета, трябва да се регистрира в регистъра на продажбите. Една продажба има следния вид (подобно на касовите бележки в супермар­кетите):

* дата и час на продажбата (например 12.06.2010, 17:45)
* списък от закупените стоки и количества (изваждат се от наличностите):
  + продукт (напр. "дини от Любимец")
  + количество (например 3.47 кг)
  + сума платена от клиента (изчислява се по текущата цена на продукта)
* обща платена сума от клиента (прибавя се в касата)

1. Продажби могат да извършват всички потребители (касиери).
2. Не е позволено да се продава стока, която не е налична в системата.
3. След като веднъж са извършени, продажби не могат да се изтриват. Ако е необходимо да се редактира или анулира продажба, може да се извърши нова продажба с отрицателно количество на закупените стоки.
4. Да се обърне внимание на факта, че е възможно е цените на зареждане за един продукт да са различни при различните зареждания на стока, но продажната цена в даден конкретен момент за всеки продукт е фиксирана.
5. Да се обърне внимание на ситуацията, при която няколко клиента паралелно пазаруват при различни касиери в супермаркета и да се внимава да не се допуска объркване на сметките при засичане на продажбите. Например, ако в супермаркета има само 3 кг. домати, не е възможно едновременно два потребителя да закупят по 2 кг. Който потребител пръв закупи свършващата стока, печели, а другият получава съобщение за недостиг на наличност.

## Зареждане на стока

1. Всяко зареждане на стока, трябва да се регистрира в списък на зарежданията. За всяко зареждане в системата постъпва следната информация:

* дата и час на зареждането (напр. 10.06.2010, 08:45)
* списък от заредените стоки и количества (прибавят се към наличностите):
  + продукт (например "хляб Добружда")
  + количество (например 150 броя)
  + цена на зареждане (ако цената на зареждане е по-висока от продажната цена, да се изведе предупреждение)
* обща платена към доставчика сума (изважда се от касата)

1. Не е позволено да се зарежда стока за повече пари, отколкото има в касата.
2. Зареждане на стока могат да извършват всички потребители (касиери).
3. Да се внимава да не се допуска объркване на сметките при засичане на операцията по зареждане на стока с паралелни продажби в същия момент.

## Наличности

За супермаркета трябва да се поддържа списък с наличностите:

* продукт (напр. дини от Любимец)
* количество (напр. 60 - подразбира се мерна единица "килограм" – тя зависи от продукта)

## Справки

1. Да се реализира справка за наличност – да отпечатва всички стоки и техните наличности към текущия момент (продукт, категория, наличност, единична продажна цена и обща продажна цена) заедно със сумарната цена на наличната стока.
2. Да се реализира предходната справка с възможност за избор на категория, за която да се извърши справката.
3. Да се реализира справка за наличност към дадена дата, която се избира от потребителя. Да се извеждат единствено продукт, категория и наличност (без цени).
4. Справка за продажби за даден период – показва всички продажби за даден период между две дати. По подразбиране да избира последните 7 дни. Да се извеждат всички продадени продукти заедно с продадените количества и постъпилите приходи за всеки продукт.

За всички таблици, извеждани в справките да се имплементира възможност за сортиране (по всяка от колоните) и страниране (извеждане на страници от по 15 реда с възможност за навигация между страниците).

## Допълнителни изисквания

* Потребителският интерфейс на системата може да е на български или на английски език (по ваш избор).
* Използване на **.NET Framework 4.0 или 4.5** и езика **C#**.
* Използване на **ASP.NET MVC 4**.
* Съхранение на данните в релационна база данни **Microsoft SQL Server 2008 или 2012**.
* Използване на **ADO.NET Entity Framework** за достъп до базата данни.
* Използване на **Microsoft Visual Studio 2010 или 2012**.
* За управлението на потребителите, регистрация, login и logout е разрешено да се използва **ASP.NET Membership**, но не е задължително изискване.
* Потребителското име на всеки трябва да бъде уникална поредица от латински букви и цифри с дължина минимум 3 символа и да започва с буква.
* Паролата трябва да бъде с дължина поне 3 символа и е задължителна за всички регистрирани потребители.
* Email адресът е задължителен и трябва да се валидира минимум по следния формат: **<знак/знаци>@<знак/знаци>.<знак/знаци>**.

## Бонуси

Допълнителни точки ще се дават за:

* Качествен програмен код – според всички препоръки от курса High-Quality Programming Code (<http://codecourse.telerik.com>).
* Наличие на unit тестове.
* Реализирана добра ползваемост – удобен, лесен за употреба и интуитивен потребителски интерфейс.
* Коректност и правилна работа при необичайни ситуации: например при въвеждане на невалидни данни (празни полета, прекалено дълги полета, невалиден формат).
* Липса на пробиви в сигурността: SQL injection, cross-site scripting, unauthorized directory traversal и други.
* Добър и удобен външен вид (уеб дизайн) – можете да ползвате ваш собствен дизайн или да адаптирате дизайн със свободен лиценз от сайтове като: [http://www.oswd.org](http://www.oswd.org/) и <http://www.openwebdesign.org>.
* Реализирането на функционалност, която не се изисква в условията на проектите не носи непременно допълнителни точки. По-важно е да се реализира коректно изискваната функционалност.

## Предаване на проектите

Проектите трябва да съдържат следните активи, пакетирани в ZIP / RAR архив:

* База данни – релационна схема на базата данни в SQL Server с примерни тестови данни в таблиците. Базата данни да се предостави като SQL скрипт, който създава релационната схема и вмъква примерни данни в таблиците. Примерните данни трябва да са смислени, не "dsafdshga", "dsagfdsa" и "test1".
* Работещо ASP.NET MVC уеб приложение, реализиращо в пълнота и според изискванията цялата функционалност на настоящия проект.
* Пълен сорс код на проекта (C# класове, модели, изгледи, контролери, страници, картинки, скриптове и други ресурси).
* По желание проектът може да се качи в AppHarbor, Windows Azure, UhuruCloud или на друг обществен облак за .NET приложения с цел по-лесна демонстрация по време на защитата.

Проектите се предават онлайн от адрес: <http://telerikacademy.com/Courses/Courses/Details/15>.

Краен срок за предаване на проектите: **10 февруари 2013 г.**

## Защита на проектите

Защитата на проектите ще се извърши в края на курса. На защитата всеки студент ще трябва да представи проекта си, да покаже каква част от изискванията са реализирани и работят и да разкаже как е реализирал функционалността, след което ще му бъде дадена задача – да реализира допълнителна функция към проекта, на място в изпитната зала.

Силно препоръчително е всеки да си носи лаптоп с подготвена работна среда, на който да работи по време на защитата.

За да спестите време по време на защитата моля подгответе достатъчно примерни данни в базата данни на вашата система, с които да покажете всичките й функции в действие. Например, за да покажете страниране на таблиците, е необходимо да имате подготвени достатъчно данни в съответната таблица.

Датата за защита на проектите ще бъде обявена по-късно.

## Дискусионна група

За въпроси и коментари може да използвате дискусионната група на курса:

* <http://forums.academy.telerik.com/asp-net>