# Автомобилен сервиз – курсов проект по „Уеб приложения с ASP.NET MVC” – вариант #5

Да се проектира и реализира софтуер за управление на автомобилен сервиз. В сервиза ежедневно се ремонтират автомобили и се издават ремонтни карти, които съдържат информация за извършения ремонт, вложените резервни части и труд. За един автомобил може да има множество ремонтни карти като всяка една отговаря на извършен ремонт. Системата трябва да предоставя различни справки, свързани с извършените ремонти.

## Потребители

1. Потребители на системата са операторите, които приемат клиентите в сервиза и попълват информация в системата. Тази информация е споделена между всички потребители, но право на редакция на дадена ремонтна карта има само потребителят, който я е създал.
2. За всеки потребител в системата се пази следната информация:

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

1. Специален потребител с username "**admin**" и предварително зададена в системата парола може да добавя, редактира и изтрива потребители. При изтриване потребителите се деактивират, но остават в системата (за да не се повреждат справките).
2. Достъп до системата имат единствено регистрирани потребители (след login) и администра­торът.

## Автомобили

1. За всеки автомобил се поддържа следната информация:

* държавен регистрационен номер (въвежда се задължително, например)
* марка
* модел
* година на производство
* номер на рама – въвежда се задължително (трябва да е уникален в системата)
* номер на двигател – въвежда се задължително
* цвят
* работен обем на двигателя (напр. 500 куб. см.)
* описание (за друга информация)
* име на собственик
* телефон за контакт със собственика

1. За автомобилите се поддържат следните операции (могат да се извършват от всеки оператор):

* Добавяне на автомобил – въвежда се и се съхранява информацията за автомобила.
* Редактиране на автомобил – позволява промяна на информацията за даден автомобил.

## Резервни части

1. Резервните части имат:

* идентификатор (уникален код по каталог)
* име (текстово наименование, примерно "спирачен диск")
* цена

1. Видовете резервни части, които могат да се ползват при ремонт на автомобили се дефинират от администратора. При създаване или редакция на ремонтна карта се използват така дефини­раните вече резервни части.
2. За резервните части се поддържат следните операции (достъпни само за администратора):

* Добавяне на резервна част.
* Редактиране на резервна част – позволява промяна на наименованието или цената на съществуваща в системата резервна част.
* Изтриване на резервна част – изтрива вече дефинирана резервна част. Изтриването прави частта неактивна и така тя може да участва в справките, но не е достъпна при нови ремонти.

## Служители в сервиза (монтьори)

Системата трябва да поддържа списък със служителите в сервиза (монтьорите, които извършват ремонтите). Всеки служител има име и фамилия. Администраторът може да добавя / редактира / изтрива монтьори.

## Ремонтни карти

1. В ремонтните карти се съдържа следната информация:

* номер на ремонтната карта
* дата на приемане на автомобила в сервиза (попълва се автоматично с текущата дата)
* дата на излизане на автомобила от сервиза (попълва се автоматично при приключване на ремонта)
* автомобил, който е се ремонтира (избира се измежду въведените в системата автомобили с подходящо търсене, например по част от рамата или от регистрационния номер)
* описание на ремонта, който е направен (списък от извършени дейности)
* име на служител, отговорен за ремонта (име на монтьора, избрано от списък)
* списък на вложените части (избират се от предварително дефинираните резервни части)
* цена на вложените части (смята се автоматично от системата)
* обща цена на извършения ремонт (въвежда се от потребителя на системата и не може да е по-малко от цената на вложените части)

1. Върху ремонтните карти се поддържат следните операции:

* Създаване на нова ремонтна карта – попълва се и се съхранява първоначалната информация за ремонтната карта. Някои полета не могат да бъдат попълнени докато ремонтът не приключи.
* Търсене на ремонтна карта (по рама или регистрационен номер на автомобил).
* Редактиране на съществуваща ремонтна карта – позволява промяна на информацията съдържаща се в нея. Не може да се редактират номер на ремонтната карта и дата на приемане на автомобила.
* Приключване на ремонт – попълва се цената на ремонта и се въвежда датата на излизане на автомобила.

1. Превключване на ремонта – ремонтната карта се издава при приемане на автомобила и се попълва по време на извършването на ремонта. Той може да отнеме повече от един ден. При приключване на ремонта се въвежда дата на излизане на автомобила от сервиза и ремонта се счита за приключен.

## Справки

1. Системата трябва да предоставя възможност за извършване на справки за незавършените ремонти. При такава справка се възвеждат следните критерии:

* начална дата (на която е приет автомобила)
* номер на рама или регистрационен номер (може и част от тях – например при въвеждане на "CA79" може да се намери автомобил с регистрационен номер "CA7972KK" или с рама номер "BMW87326CA793").

1. Системата трябва да предоставя възможност за извършване на справки за завършените ремонти в даден период между две дати и паричните постъпления от тях.

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: **27 януари 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>