# Система за отчитане на работното време – курсов проект по „Уеб приложения с ASP.NET MVC” – вариант #4

Да се проектира и реализира информационна система за отчитане на работното време на служителите в дадена фирма. Системата трябва да поддържа списък със задачите и броя часове, които всеки служител е работил по всяка от тях. Всеки служител сам отчита в системата работата си по съответните задачи. Администраторът управлява служителите и задачите.

## Видове потребители

1. В системата има три вида потребители:

* Посетители (без автентикация) – те могат единствено да влязат (login) в системата.
* Регистрирани потребители (служители) – те могат да влизат в системата (login), попълват своите работни карти и да правят справки.
* Администратор – той е специален потребител, който може да създава потребители, да управлява задачите и да прави справки. Администраторът е потребител с username "**admin**" и парола зададена предварително в системата.

## Служители

1. За всеки служител (потребител в системата) трябва да се пази следната информация:

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

1. Служителите могат да бъдат създавани, изтривани и редактирани единствено от админи­стратора на системата. Позволена е редакция на всички данни за потребителя с изключение на потребителското име. При изтриване на потребител той се запазва, но става неактивен (така той се появява в справките, но не може да взаимодейства със системата).

## Задачи

1. За всяка задача в системата се съхранява следната информация:

* Наименование
* Кратко описание
* Дата на започване на задачата (незадължителна)
* Дата на завършване на задачата (незадължителна)
* Изпълнители – множество от потребители, които могат да работят по задачата
* Общ брой човекочасове предвидени за изпълнение на задачата – изчисляват се автоматично по формулата: брой изпълнители \* 8 часа \* брой работни дни за изпълнение. Може да се редактира ръчно, незадължително поле.
* Изразходвани до момента човекочасове (смятат се автоматично).
* Статус на задачата – една задача може да е **нова**, **започната**, **завършена** и **отказана**.

1. Задачите могат да бъдат създавани, изтривани и редактирани единствено от администратора на системата. При изтриване на задача тя се запазва, но става неактивна (така тя се появява в справките, но не може да се използва повече при отчитане на работни карти).

## Работни карти

1. След като един служител е работил по някаква работа, той попълва работна карта съдържаща следната информация:

* начало на работата (дата и час, по подразбиране текущата дата и час)
* продължителност (часове и минути, максимум 14 часа)
* задача, по която е работил (избира се от списъка със задачите за съответния потребител)
* кратко описание на извършеното действие (задължително се попълва)

1. Веднъж попълнил работна карта служителят не може я променя или изтрива.
2. Единствено администраторът може да редактира и изтрива работни карти.

## Справки

Системата трябва да поддържа следните справки, достъпни за всички потребители (включително и за администратора):

1. Извеждане на таблица с всички задачи (име, начало, край, брой изпълнители, планирани часове, изработени до момента часове, статус).
2. Извеждане на информация за дадена задача – таблица със служителите, които са работели по задачата и времето, което всеки служител е отделил на дадена задача.
3. Извеждане на всички служители, броя задачи, по които е работил, и общ брой изработени часове.
4. Извеждане на информация за избран служител – задачи, по които е работил служителят и брой часове за всяка задача.

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>