Требуется создать онлайн-менеджер фотографий с удобной навигацией, возможностью поиска фотоснимков, управления фотоальбомами. К управлению фотографиями допускаются только зарегистрированные пользователи. Любые пользователи могут просматривать снимки из виртуальных альбомов.

Основные странички сайта с примерным дизайн-макетом:

* + Домашняя страница (содержит список созданных пользователем альбомов; навигационные ссылки и кнопки; поиск по ключевым словам; расширенный поиск - поиск по свойствам фотографий):

Страница управления фотографиями (добавление/удаление фотографий, переход на страницу создания фотоальбома и на страницу редактирования свойств снимка):

* Страница свойств выбранной фотографии;
* Страница управления фотоальбомами;
* Публичная страница фотоальбома;
* Страница входа на сайт и регистрации.

**Функциональные требования**

* 1. Любые пользователи могут просматривать любые альбомы по прямым ссылкам.  
       
     // Задачка со звездочкой:Любые пользователи, не являющиеся хозяевами альбома могут выставлять Like/Dislike на фотографиях. При этом число + и - должно отображаться.
  2. Из любых фотографий можно создать виртуальный фотоальбом с названием и кратким описанием. При этом одна и та же фотография может быть включена в несколько фотоальбомов без ограничений. При удалении фотки она должна пропадать и из всех альбомов, в которые была включена.
  3. Название альбома должно быть уникальным. По названию альбома будет генерироваться публичная ссылка на этот альбом.
  4. Для альбома нужно сделать кнопку - Get Link. При нажатии на кнопку система выдает уникальный адрес альбома с URL shortening'ом (наподобие [http://www.myphotosite.com/albums/Genre-Album где 'Genre Album'](http://www.myphotosite.com/albums/Genre-Album%20%D0%B3%D0%B4%D0%B5%20%27Genre%20Album%27" \t "_blank) - название фотоальбома)
  5. На странице редактирования фотографии нужно сделать возможность ввода таких полей:
     + Название
     + Дата и время съемки
     + Место, где был сделан снимок
     + Модель фотоаппарата
     + Фокусное расстояние объектива
     + Диафрагма
     + Скорость затвора
     + ISO
     + Использовалась ли вспышка

По этим же полям нужно сделать расширенный поиск.

// Задачка со звездочкой: зачитывать мета-данные при загрузке фото.

* 1. Пользователи сайта бывают двух видов: обычные и платники. Обычные пользователи могут загрузить не более 30 снимков и создать не более 5 альбомов. При достижении лимита нужно отображать сообщение и блокировать соответствующие кнопки. У коммерческих пользователей никаких ограничений нет.
  2. Способ навигации между фотографиями в альбоме остается на усмотрение выполняющего задание. Как вариант: текущая карточка показывается в увеличенном виде, а остальные в виде иконок.

**Требования к архитектуре приложения**

1. Платформа: ASP.NET MVC 4+; ADO.NET Entity Framework 4+
2. Использовать стандартную трехуровневую архитектуру.
3. Использовать технологию Ajax, использовать jQuery + MVC Web API.
4. Все данные при вводе должны валидироваться.
5. При загрузке на сайт фотографий сделать проверки:
   * ограничение на объем (фото должно весить не более 500К);
   * проверять MIME-type (принимать только картинки-JPEG);
   * на серверной стороне убеждаться, что картинка - действительно JPEG.
6. Использовать строготипизированные View.
7. Написать свои custom helpers.
8. Использовать фильтры.
9. JS-скрипты и CSS-файлы должны подгружаться с помощью bundles. Включить сжатие контента.
10. Поиск по картинкам должен осуществляться с помощью stored procedure, принимающей на вход keyword, и возвращающей список айтемов, удовлетворяющих этому критерию. Расширенный поиск по картинкам можно реализовать иначе. Например, с помощью query object паттерна.
11. Все ошибки и ворнинги должны логироваться в БД. Пользователь не должен видеть страничек с 500 server error. Требуется отображать свою страничку с добрым мессаджем и не нервировать пользователя : )
12. Сайт не должен падать при сабмите тэгов в полях ввода.
13. Обычный и платный пользователи должны наследоваться от базового класса Пользователь. Сделать это нужно с пом. Entity Framework. Можно использовать любой из доступных видов наследования.
14. Картинки хранить в базе.
15. Сделать возможность загрузки на UI картинок в трех размерах:
    * **миниатюра** - отображается на странице альбома и на всех промежуточных страницах;
    * **средний размер** - для текущей картинки в альбоме;
    * **1 к 1** - открывается в новом окне при клике по картинке.
16. В случае выбора Lazy loading или отказа от нее обосновать архитектурное решение.

**Нефункциональные требования**

* Сайт не должен падать после перезагрузки SQL сервера.
* Пользовательская сессия не должна рваться при перезапуске Пула приложений вебсервера (IIS Application Pool).
* Сайт должен быть устойчив к атакам SQL injection, Cross-site scripting и Session hijacking.
* При выполнении задания, пожалуйста, используйте только релизные версии фреймворков и библиотек. Использование alpha/beta/rc\_x и прочих сырых и найтли-билд продуктов возбраняется и проверяющий в праве попросить переделать проект.
* Также просьба в выполненное задание включать полный дамп БД с тестовыми данными либо SQL скрипты создающие схему данных и добавляющие тестовые данные в базу.

# Рекомендуемые ссылки

Базовые знания о перечисленных web атаках можно найти <http://www.darkreading.com/security/news/208803744>здесь

Информацию по технологиям можно найти тут:

\\portal2\PortalLibrary\KB\.NET\ASP.NET\  
\\portal2\PortalLibrary\KB\.NET\ASP.NET MVC\

<http://www.asp.net/>  
<http://www.asp.net/MVC>

Entity Framework on MSDN

По мере готовности или при возникновении любых вопросов пишите нам: [training.ASP.NET](mailto:training.ASP.NET@dataart.com)

Спасибо, что вы с нами.