



Тестовое задание #4 {ASP.NET MVC}



Тестовое задание для оценки кандидатов на соответствие должности Разработчика ПО 1-го уровня (Software Developer 1st Level).

Profit Labs Ltd.
St. Petersburg, Russia
+7.812.493.2567
www.profit-labs.ru

Содержание


СОДЕРЖАНИЕ	2
1 ОПИСАНИЕ ЗАДАЧИ	3
1.1 Вход в СИСТЕМУ	4
1.2 СОЗДАНИЕ УЧЕТНОЙ ЗАПИСИ ПОЛЬЗОВАТЕЛЯ	4
1.3 РЕДАКТИРОВАНИЕ УЧЕТНОЙ ЗАПИСИ ПОЛЬЗОВАТЕЛЯ	5
1.4 УДАЛЕНИЕ УЧЕТНОЙ ЗАПИСИ ПОЛЬЗОВАТЕЛЯ	5
1.5 СОЗДАНИЕ СООБЩЕНИЯ.....	5
1.6 ПРОСМОТР СПИСКА СООБЩЕНИЙ	6
1.7 ПРОСМОТР СООБЩЕНИЯ	6
1.8 УДАЛЕНИЕ СООБЩЕНИЯ.....	6
1.9 ПРОСМОТР СПИСКА ПОЛЬЗОВАТЕЛЕЙ	6
1.10 ВЫХОД ИЗ СИСТЕМЫ	6
1.11 ПРИМЕР ЦЕЛЕВОГО ИСПОЛЬЗОВАНИЯ.....	7
2 ТРЕБОВАНИЯ К СЛОЮ БАЗЫ ДАННЫХ.....	8
3 ТРЕБОВАНИЯ К СЛОЮ ДОСТУПА К ДАННЫМ.....	9
4 ТРЕБОВАНИЯ К СЛОЮ БИЗНЕС ЛОГИКИ	10
5 ТРЕБОВАНИЯ К СЛОЮ ПРЕДСТАВЛЕНИЯ.....	11
6 ТРЕБОВАНИЯ К СЛОЮ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА	13
7 ОБЩИЕ ПОЖЕЛАНИЯ	14
8 КОНТАКТНАЯ ИНФОРМАЦИЯ.....	15

1 Описание задачи

Необходимо разработать небольшое web-приложение для обмена сообщениями между пользователями. Приложение должно предоставлять следующие возможности:

1. Первоначальная регистрация личной учетной записи пользователя {User} (создание учетной записи с обязательными полями почтового адреса {Email} и пароля {Password}, и необязательными — имя {FirstName}, фамилия {LastName} и телефон {Phone}).
2. Вход в систему с учетной записью пользователя (используя поля из п.1).
3. Управление учетной записью пользователя (редактирование / удаление личной учетной записи пользователя).
4. Просмотр списков Входящих {Received или Inbox} и Исходящих {Sent} сообщений {Message} пользователя (в отдельной области основного экрана просмотр того или иного списка сообщений как набор элементов с заголовком {Subject} и датой {Date}) упорядоченных в порядке убывания по дате, и, удаление каждого отдельного сообщения по требованию (посредством нажатия кнопки/иконки/гиперссылки в области элемента сообщения).
5. Открытие сообщения на прочтение (в отдельной области основного экрана открытие сообщения только для чтения с представленными полями отправитель {From}, получатель {To}, заголовок {Subject} и самим телом сообщения {Body}).
6. Создание нового сообщения (в отдельной форме или области основного экрана открытие нового (=пустого) сообщения на редактирование с полями из п.5).
7. Ответ на Входящее сообщение (в текущем сообщении инициируется посредством нажатия на кнопку/иконку/гиперссылку Ответить {Reply}; при этом создается новое сообщение с пустым или предыдущим телом сообщения, но обязательно сохраняется заголовок письма (возможно с приставкой НА: {Re:}), и автоматически подставляется From и To поля соответствующим образом).
8. Просмотр списка зарегистрированных пользователей (в отдельном окне просмотр списка пользователей, зарегистрированных в системе; это окно также должно быть доступно при выборе получателя в процессе создания нового сообщения).

В приложении не предполагается использовать роли пользователей, т.к. она единственная, но есть требования к авторизации прав в системе:

	<ul style="list-style-type: none"> ✓ Только сам пользователь может изменять и удалять свою учетную запись. ✓ Пользователь может посылать сообщения только от своего имени. ✓ Пользователь может просматривать только свои списки сообщений (=почтовый ящик), но не других пользователей.
---	---

Учетная запись пользователя должна иметь следующий минимальный набор характеристик:

- ☐ Email (уникальное в рамках всего приложения; обязательное поле)
- ☐ Password (вариант хранения — Hashed; обязательное поле)
- ☐ FirstName (имя; необязательное поле — если не указано, в системе показывается Email, иначе комбинация FirstName + LastName)
- ☐ LastName (фамилия; необязательное поле — если не указано, в системе показывается Email, иначе комбинация FirstName + LastName)

- ☐ Phone (телефон; необязательное поле – в системе показывается только в списке пользователей)

Сообщение должно иметь следующий минимальный набор характеристик:

- ☐ From (отправитель; обязательное поле)
☐ To (получатель; обязательное поле)
☐ Subject (заголовок; обязательное поле)
☐ Body (тело; необязательное поле)

Далее представлены необходимые для реализации варианты использования приложения (Use Cases).


1.1 Вход в систему

Начальное условие :	<p>Запущено приложение. Перед пользователем окно входа в систему (Login) со следующими полями:</p> <ul style="list-style-type: none"> ▪ Почтовый адрес пользователя {Email} – текстовое поле. ▪ Пароль {Password} – текстовое поле для ввода пароля. <p>Также присутствует кнопка Вход (Login / Sign In) и гиперссылка для перехода на экран регистрации нового пользователя.</p>
Действие :	<ol style="list-style-type: none"> 1. Пользователь вводит Email; 2. Вводит Password; 3. Нажимает Вход.
Ожидаемый результат :	<ul style="list-style-type: none"> ➤ Если данные правильные и аутентификация прошла, тогда пользователь попадает на основной экран приложения. ➤ Если данные неверны или по какой-то причине вход в систему был неуспешен, тогда выдается сообщение об ошибке с соответствующим описанием.

1.2 Создание учетной записи пользователя


Начальное условие :	<p>В приложении пользователь находится на экране регистрации нового пользователя. На форме присутствуют все поля учетной записи пользователя и кнопка Sign Up.</p> <p>Также есть гиперссылка возврата на страницу входа в систему.</p>
Действие :	<ol style="list-style-type: none"> 1. Пользователь заполняет все обязательные и по желанию необязательные поля учетной записи. 2. Нажимает кнопку Sign Up.
Ожидаемый результат :	<ul style="list-style-type: none"> ➤ Если данные в полях указанные некорректно (например, не заполнено обязательное поле, введен некорректный телефонный номер или введено уже используемый в системе почтовый адрес), тогда стандартными средствами валидации пользователя информируют о некорректности введенного значения. ➤ Если все поля заполнены корректно, то происходит попытка сохранения нового пользователя в БД. При этом если сохранение выполнено корректно, тогда пользователя автоматически переводят на основной экран приложения, где он видит пустой список сообщений и кнопки просмотра пользователей системы

	и создания нового сообщения. Если же при сохранении произошла исключительная ситуация, тогда пользователю показывается соответствующее сообщение об ошибке.
--	---

	При вводе пароля необходимо спрашивать его подтверждение (2 поля).
---	--

1.3 Редактирование учетной записи пользователя

Предполагается самостоятельная проработка этого случая кандидатом.
Единственное требование:

	Пользователь не может менять поле Email после создания учетной записи, только остальные поля могут быть изменены.
---	---

1.4 Удаление учетной записи пользователя

Предполагается самостоятельная проработка этого случая кандидатом.

1.5 Создание сообщения

Начальное условие:	В приложении пользователь находится на его основном экране. В основном меню приложения пользователь нажимает кнопку создания нового сообщения {Create}. В отдельном окне или соответствующей области основного экрана приложения открывается форма нового (=пустого) сообщения с его полями.
Действие:	<ol style="list-style-type: none"> 1. Пользователь заполняет обязательные и по желанию необязательные поля сообщения. 2. Нажимает кнопку Послать {Send} или Отмена {Cancel}.
Ожидаемый результат:	<ul style="list-style-type: none"> ➤ Если пользователь нажал Отмена, тогда форма закрывается, и новое сообщение не отправляется. ➤ Если пользователь нажал Послать и при этом данные в полях указанные некорректно (например, не указано обязательное поле), тогда стандартными средствами валидации пользователя информируют о некорректности введенного значения. ➤ Если пользователь нажал Послать и все поля заполнены корректно, то происходит попытка отсылки сообщения адресату. В случае успеха окно закрывается или область основного экрана меняет свое состояние на исходное, и, при этом текущий пользователь видит это сообщение в списке Исходящих {Sent}, а получатель видит его в списке Входящих {Received или Inbox}. Если же при сохранении произошла исключительная ситуация, тогда

	пользователю показывается соответствующее сообщение об ошибке, но форма не закрывается.
--	---

1.6 Просмотр списка сообщений

Предполагается самостоятельная проработка этого случая кандидатом.
Единственное требование:



	Должна быть предоставлена возможность отдельного просмотра Входящих и Исходящих сообщений.
--	--

1.7 Просмотр сообщения

Предполагается самостоятельная проработка этого случая кандидатом.
Единственное требование:



	В режиме просмотра сообщения его поля не могут быть отредактированы пользователем.
--	--

1.8 Удаление сообщения

Предполагается самостоятельная проработка этого случая кандидатом.
Единственное требование:



	Пользователь может удалять сообщения только в своих списках сообщений (=почтовом ящике), не изменяя состояние почтовых ящиков своих отправителей и получателей.
--	---

1.9 Просмотр списка пользователей

Предполагается самостоятельная проработка этого случая кандидатом.

1.10 Выход из системы

Предполагается самостоятельная проработка этого случая кандидатом.

1.11 Пример целевого использования

1. Пользователь Отправитель вошел в систему.
2. Создал сообщение и послал его.
3. Нашел его первым в списке Исходящих сообщений, открыл (на чтение, например для проверки корректности смысла сообщения).
4. Вышел из системы.
5. Пользователь Получатель вошел в систему.
6. Открыл (раскрыл) список входящих сообщений и нашел его.
7. Открыл его (на чтение).
8. Нажал Ответить {Reply}, заполнил тело сообщения своим текстом и нажал Послать {Send} ответ.
9. Вышел из системы.
10. Отправитель снова зашел в систему и увидел ответ в списке Входящих сообщений.
11. Открыл сообщение и прочитал ответ Получателя.

2 Требования к слою базы данных

1. **Рекомендация:** Для выполнения данного задания предпочтительно использовать редакцию **LocalDb** (редакция СУБД MS SQL Server 2012 для разработчиков в Visual Studio – входит по умолчанию в MS **Visual Studio 2012**).
2. Если не используется LocalDb, тогда можно использовать любую другую версию/редакцию MS SQL Server (**но не Compact**).
3. При этом схема БД либо генерируется автоматически при помощи возможностей ADO.NET Entity Framework (IDatabaseInitializer<TContext>), либо ее разработка должна быть осуществлена вручную одним из следующих вариантов:
 - ☐ Средствами Microsoft SQL Server Management Studio, тогда для проверки задания требуется прислать ее в виде набора файлов – SQL скриптов (если важен порядок исполнения скриптов, то его необходимо указать в названии файлов, например "01 Tables.sql", "02 View.sql", ...)
 - ☐ **Рекомендация:** Средствами специального типа проекта Visual Studio – SQL Server Database Project (см. Рисунок 2-1).

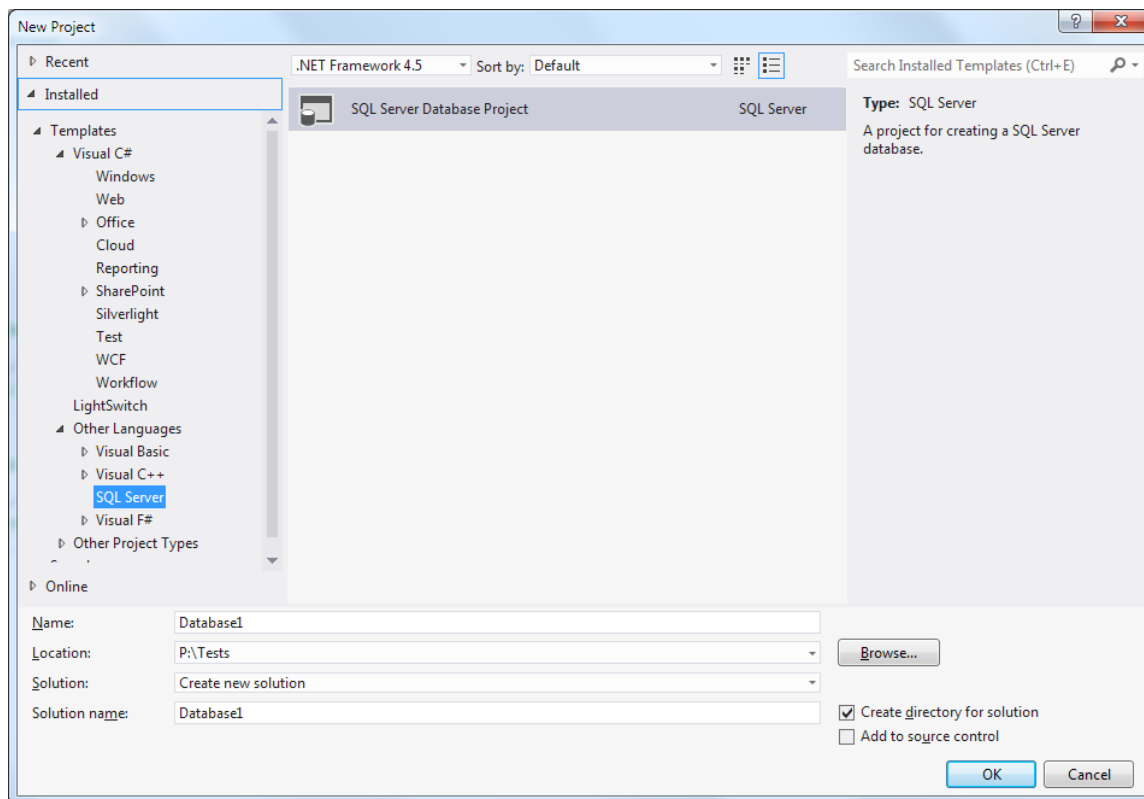


Рисунок 2-1 Тип проекта SQL Server Database Project.

3 Требования к слою доступа к данным

1. **Требование:** Необходимо разработать слой доступа при помощи технологии **ADO.NET Entity Framework 5.0 (.NET 4.5)** с подходом **Code First**.
2. **Рекомендация:** Предпочтительно выделить слой доступа к данным в отдельный проект VS.

4 Требования к слою бизнес логики

1. **Рекомендация:** Требования к наличию данного слоя в приложении не предъявляется, выделение его в отдельный проект оставляется на откуп кандидату. Однако рекомендуется логику работы с данными вынести в отдельный(е) класс(ы), в методах которого(ых) выполнять комплексную работу с данными (сохранение в нескольких таблицах одновременно в рамках единой транзакции), а также осуществлять проверку прав доступа текущего пользователя к этим данным, где это возможно/необходимо.

5 Требования к слою представления

1. **Требование:** Необходимо использовать технологию [ASP.NET MVC 4.0 \(.NET 4.5\)](#) и ее стандартные возможности (фильтры авторизации и обработки ошибок; стандартные подходы к работе с моделями данных {Model и ViewModel} и формированию представлений {View} где это возможно).
2. **Требование:** Все действия {Action} контроллеров {Controller}, отвечающие за переходы между страницами web-приложения должны быть исключительно **HttpGet**, действия же отвечающие за работу данными должны быть исключительно **HttpPost**.
3. **Требование:** Во всех действиях контроллеров, отвечающих за работу с данными, должна быть проверка, что данное действие может выполнить текущий авторизованный пользователь (например, чтобы не допустить удаление другого пользователя или сообщений другого пользователя текущим).
4. **Требование:** Все действия контроллеров, отвечающих за работу с данными, должны быть разработаны с учетом того, что они могут быть вызваны только посредством Ajax вызовов. Они не должны возвращать готовые HTML представления (View или Partial View), но исключительно данные, например – в формате JSON.
5. **Рекомендация:** При создании ASP.NET MVC проекта рекомендуется использовать вариант шаблона **Empty** (см. Рисунок 5-1) и разрабатывать его с нуля, однако можно подсматривать какие-то примеры реализации отдельных частей веб-приложения и в других вариантах шаблона проекта.
6. **Рекомендация:** В качестве параметров действий контроллеров рекомендуется по возможности использовать объекты {Model или ViewModel}, нежели примитивные типы данных.

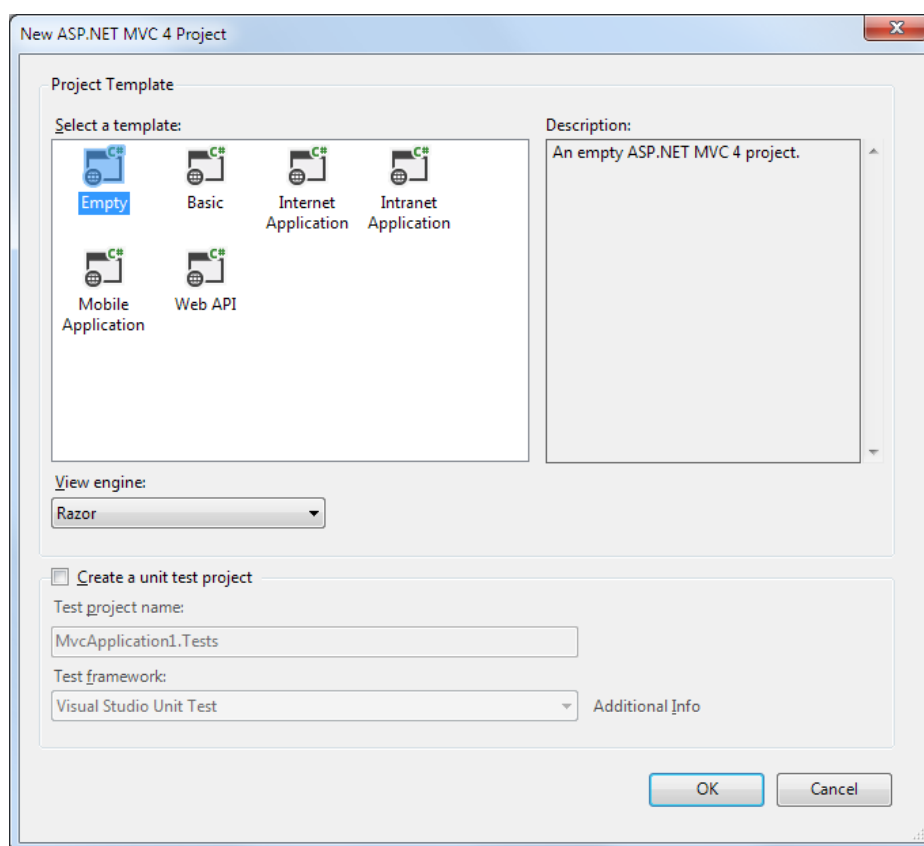


Рисунок 5-1 Варианты шаблонов проекта ASP.NET MVC 4 Web Application

6 Требования к слою графического интерфейса

1. **Требование:** Необходимо использовать следующие фреймворки:
 - . Основной – библиотека [jQuery](#).
 - а. В качестве UI фреймворка – [Twitter Bootstrap](#).
2. **Требование:** Необходимо использовать CSS. Все CSS стили должны находиться в отдельном файле, никаких inline стилей на View.
3. **Требование:** Необходимо использовать блочную верстку. Приветствуется использование HTML5 тегов, таких как header, footer, section, nav и т.д.
4. **Требование:** Для валидации форм использовать HTML5 Form Validation.
5. **Требование:** JavaScript код должен быть в отдельных js файлах, писать inline скрипты на View, только если это крайне необходимо.
6. **Требования:** Использование HTML5/CSS3, грамотное, читабельное и чистое оформление CSS стилей и JS кода.
7. **Рекомендация:** Возможность сортировки сообщений по дате, имени отправителя/получателя или теме.
8. **Рекомендация:** Постраничное отображение списка писем или бесконечная прокрутка как на [Twitter](#) или [Вконтакте](#), на выбор Кандидата.
9. **Рекомендация:** Автоматическое обновление входящих сообщений по таймеру, реализованному средствами JavaScript функции setTimeout (например, обновление списка Входящих сообщений каждые 30 сек).
10. **Рекомендация:** Если останется время, рекомендуется сделать адаптивный дизайн средствами CSS.

7 Общие пожелания

1. При выполнении задания приветствуется инициатива в выборе подходов реализации (например: использовать только стандартные ASP.NET Membership & Role providers и расширять БД дополнительными таблицами; использовать свою схему БД и реализовать свои ASP.NET Membership & Role providers; или же не использовать стандартные, а реализовать исключительно свой подход).
2. При выполнении задания также приветствуется аккуратное оформление внешнего вида приложения и самостоятельная проработка неоднозначных требований.
3. Выбор языка пользовательского интерфейса приложения на усмотрение кандидата (русский или английский; последнее приветствуется, но не обязательно).
4. Код должен быть оформлен в соответствии с установленными в компании правилами оформления (приветствуется на английском, но не обязательно).
5. В коде должен повсеместно присутствовать комментарии (как XML-Documenting, так и комментарии в теле методов).
6. Приветствуется использование Generics, Auto Property, Lambda Expressions, но только если это необходимо.
7. Присланный код должен компилироваться.
8. Присланные сценарии для БД должны исполняться без ошибок. Необходимо указывать порядок исполнения сценариев, если это важно.
9. Задание рассчитано на 2-3 недели. В независимости от того, что сделано по прошествии этого срока необходимо предоставить текущий результат.
10. Вместе с выполненным заданием необходимо также представить отчет о реально затраченных часах (или рабочих днях исходя из расчета 8 часов в день) на его выполнение.

8 Контактная информация

Метелев Алексей

Директор / Руководитель отдела развития / Менеджер проектов

Email	am@profit-labs.com
Тел.	+7.812.493.25.67
Мобильный	+7.921.309.62.24
Адрес	ООО «ПрофИТ Лабс» ул. Оптиков, 4/2 БЦ «Лахта», офис № 407 197374 Санкт Петербург Россия

Ренат Фахрутдинов

Технический директор / Руководитель отдела разработки / Менеджер проектов

Email	rf@profit-labs.com
Тел.	+7.812.493.25.67
Мобильный	+7.921.747.32.53
Адрес	ООО «ПрофИТ Лабс» ул. Оптиков, 4/2 БЦ «Лахта», офис № 407 197374 Санкт Петербург Россия