

# Тестовое задание

## Сценарий

Необходимо разработать систему сбора и просмотра контактов в CRM. Основными возможностями системы являются быстрый поиск, добавление и удаление контактов из одной общей страницы.

## Требования

1. В задании будут использоваться стандартные объекты Contact, Account, Case. Создавать новые объекты не нужно.
2. У объекта Contact необходимо добавить новое поле Contact Level (picklist) со значениями Primary, Secondary, Tertiary.

## Визуальная часть (Front End)

Визуальная часть задания представляет из себя VF страницу, на которую можно будет попасть через 'Contacts Manager' tab, ее надо будет создать. На этой странице должны отображаться все существующие контакты.

Необходимо отобразить следующие поля: Name (link), Email, Contact Level (picklist), Account (lookup), Owner (lookup), Created By (lookup), Created Date.

Таблица должна поддерживать пагинацию (по 10 контактов на странице) и сортировку по любому столбцу (по клику на заголовке).

Напротив каждого контакта нужно показать кнопку Del, которая будет удалять контакт из базы данных.

Также на этой странице должно быть поле для поиска контактов по имени (частичное совпадение по first/last name). И кнопка Search. Результаты поиска должны отображаться в той же таблице и должны показывать те же столбцы.

Кроме этого необходимо сделать кнопку 'New Contact', которая откроет новую страницу или покажет popup на той же странице, где можно будет ввести First Name, Last Name \*, Email \*, Contact Level, Account и нажать Save или Cancel.

## Серверная часть (Back End)

При создании нового контакта автоматически должен создаваться Case при помощи триггера. Case должен привязываться к новому контакту. Если контакт ссылается на аккаунт, тогда новый Case также должен ссылаться на этот аккаунт.

Поле Status у нового Case должно принимать значение 'Working', а поле Case Origin должно принимать значение 'New Contact'.

Поле Owner у Case должен быть таким же как Owner у аккаунта, на которого ссылается контакт. Также поле Priority у нового Case должно зависеть от того, какое значение было выбрано в поле Contact Level у контакта. Если оно было Primary тогда Priority = 'High', если оно было 'Secondary' тогда Priority = 'Medium', если оно было 'Tertiary' тогда Priority = 'Low'.

Весь код классов и триггеров должен быть покрыт юнит тестами. Тесты должны не только покрывать код, но также и тестировать правильность логики ваших методов.

## Ресурсы

Задание необходимо выполнить на новом Free Developer Edition Org, который можно зарегистрировать по ссылке: <https://developer.salesforce.com/signup>

Вся необходимая документация есть здесь: <https://developer.salesforce.com/docs/>

## Рекомендации

При оценке задания будет приниматься во внимание множество факторов. Это внешний вид, юзабилити, количество вопросов, полнота реализации, качество и внешний вид кода. Рекомендуем сделать внешний вид максимально похожий на стандартный внешний вид Salesforce.

По возможности информировать пользователя об критических шагах, которые он производит.

При выполнении нужно рассчитывать на то что количество контактов может быть большим, код должен нормально работать. Для этого нужно принимать во внимание рекомендации по [Best Practices](#).

При необходимости Вы можете создавать новые поля, страницы, классы, но не стоит оставлять кучи мусора, в которых потом будет сложно разобраться.

Старайтесь делать код простым и понятным.

Обязательно наличие полноценного покрытия тестами - Code Coverage (test methods).

## Отправка на проверку

Отправлять на проверку задание нужно путем передачи логина и пароля к Salesforce Org.

Перед отправкой на проверку необходимо добавить диапазон адресов 0.0.0.0 - 255.255.255.255 в Setup -> Manage Users -> Profiles -> System Administrator -> Login IP Ranges.