

ТЗ на первую версию Users (техническое задание)

Система Users, сотрудники
Делаем отдельную систему пользователей.

1. Регистрация / авторизация

Там есть регистрация, ник + пароль → заводится карточка в этой же системе.
Плюс регистрация через соцсети.

Через SOAP / REST методы:

- `doRegister` — в ответе карточка созданного клиента или текст ошибки. Пользователь создается и появляется в системе. Автор у него будет «SOAP»
- `doLogin` — проверка правильности логина-пароля. Возвращает true / false

2. Список сотрудников

Главная — список сотрудников, возможность создать нового.
Ты можешь посмотреть любого, но изменить или удалить только своего.

Доступен поиск по ФИО пользователя, его емайлу или автору (отсортировать всех, кого создал сам). Также есть фильтрация по дате изменения — посмотреть всех, созданных / измененных вчера итд. Все фильтры можно сбросить одной кнопкой.

Колонки в списке сотрудников можно сортировать. Сортировка идет по всей базе, а не только по той странице, на который ты находишься.

Если сотрудник свежесоздан / изменен, он попадает наверх списка → бай дефолт сортировка по дате изменения по убыванию.

3. Карточка сотрудника

Поля карточки, в основном просто текстовые поля:

Email

ФИО

Пол --- выпадайка

Дата рождения --- дата

Начал работать в компании --- дата
Хобби
имя1
фамилия1
отчество1
Кошечка
Собачка
Попугайчик
Морская свинка
Хомячок
Белочка
Телефон
Адрес

Создавать дубли по нику или ФИО нельзя, мы считаем, что все ФИО уникальны.

SOAP / REST методы:

- `getUser` — возвращает карточку пользователя;
- `UpdateUserOneField` — изменить одно любое поле у пользователя;
- `FullUpdateUser` — изменить все поля. В запросе обязательно передать все. Если оставить пустым — поля затрутсся

4. Список задач

Говорят, нынче модно писать реализацию туду-листов, вот и мы напишем =)
На самом деле это ради обертки над `сron` (см след пункт).

Заходишь в раздел, можешь создать задачу:

- Название `string(50)`
- Описание `string(500)`
- Ответственный (тут отображаются ФИО всех сотрудников)

Она появляется в списке задач, где можно ее изменить, удалить или добавить в расписание.

Простой пользователь видит только свои задачи, менеджер → вообще все. И все может менять.

SOAP / REST методы:

- `CreateTask` — создать задачу, в ответе сама задача. Она появляется в интерфейсе.

- UpdateTask — обновить задачу. В ответе будет просто «Задача успешно изменена!»
- DeleteTask — удалить задачу из списка. Вернет текст «Задача с ID 17 успешно удалена». Чтобы удалить задачу, нужно знать ее ID. ID возвращается в ответе метода CreateTask + его видно в режиме просмотра / редактирования задачи в интерфейсе.
- AddTaskInCron — добавить в расписание. Ответ будет, что задача добавлена в расписание + дата следующего запуска. Например, «*Расписание успешно добавлено в задачу 18 Следующая дата запуска 06.04.2017 12:00*»

5. Уведомления

Сделать уведомления о задаче:

Я создала задачу, повесив на пользователя А. Задача должна сработать через 30 минут.

1. У пользователя сразу же появляется уведомлялка — ну как [Gmail Checker](#), типа «на вас повесили задачу».
2. За 5 минут до начала задачи приходит вторая уведомлялка «Через 5 минут будет задача такая-то».
3. Ну и ровно в срок уведомлялка.
4. Когда открываешь уведомлялку (это сейчас в блоке “0 заданий / 1 задание...”) должна быть кнопка “сделал”, только тогда она пропадает из расписания. Плюс кнопка «продолбал» :) Нажатия на эти кнопки логируются: “Пользователь А подтвердил, что сделал задачу (название задачи) / Пользователь А продолбал задачу (название задачи)”

Выполненные и пропущенные задачи должна быть возможность просмотреть --- Поля из самой задачи + кто и во сколько ее создал и когда она была в расписании исходно. И когда попала в "выполненные" / продолбанные.

На вкладке <http://users.bugred.ru/tasks/my/completes> колонки "Создана" и "Выполнена" (в итоге сделали одну, «время» выполнения)

На вкладке <http://users.bugred.ru/tasks/my/fails> такая же колонка со временем.

**

Сценарий (применялся, когда разработчик запутался в моих требованиях)

У тебя есть список задач:

1. Одноразовые (свозить кота на кастрацию)
2. Многоразовые (поливать цветы 2 раза в неделю)

В списке задач ты видишь актуальные. Кастрацию пока не сделал? Висит и мозолит глаза без даты. Как тока свозил или решил "пущай живет", то есть нажал на кнопку — задача больше не мельтешит перед глазами.

Многоразовые — ты вчера не полил цветок, нажал "продолбал", увидел следующую дату поливки по крону)

Все твои телодвижения логируются: изменил задачу, поставил в расписание, отклонил, сделал
а менеджер может за тобой следить)

**

Оповещения — <http://users.bugred.ru/tasks/notify/index.html>.

Ты видишь только свои, менеджер видит все. Он может их фильтровать по конкретному человеку и выставлять диапазон проверки.

6. Менеджер

Отдельная роль, настраивается для любой учетки в админке. В бекап вшит один менеджер: manager@mail.ru / 1.

Менеджер может наблюдать за сотрудниками:

- Он видит все оповещения + может их сортировать и фильтровать;
- Он видит все текущие задачи, все продолбанные и все выполненные;
- Он может отредактировать любую задачу;