# Задание на создание Личного кабинета посетителя (ЛК)

**Краткое описание задач:**

1. Создать файл миграций для БД на основе представленной структуры
2. Создать обработку роутов для перечня страниц
3. Создать простые шаблоны для страниц
4. Создать соответствующие методы обработки в контроллере для созданных страниц и обработки отправки форм

Авторизация должна работать не на стандартном движке авторизации Laravel, а с учетом что планируется три разных группы пользователей и для каждой будет использоваться отдельная таблица. Валидация доступа должна осуществляться только для страниц http://servername.ru/my/ и глубже.

**Общее описание функционала**

ЛК представляет собой закрытую зону для каждого из пользователей с доступом по авторизации. Регистрация пользователей осуществляется через соответствующую форму с подтверждением через почту. В ЛК пользователю показываются выбранные им мероприятия и экспоненты, а также информационные сообщения о мероприятии.

Оформление страниц минимально необходимое, для обработки ajax запросов используется jquery.

1. **Форма регистрации**

Форма регистрации включает в себя два типа полей (поля прямого доступа в БД) и поля в формате JSON.

Прямыми полями являются:

Фамилия, Имя, Отчество, емейл, код регистрации, статус активации

Поля формата JSON:

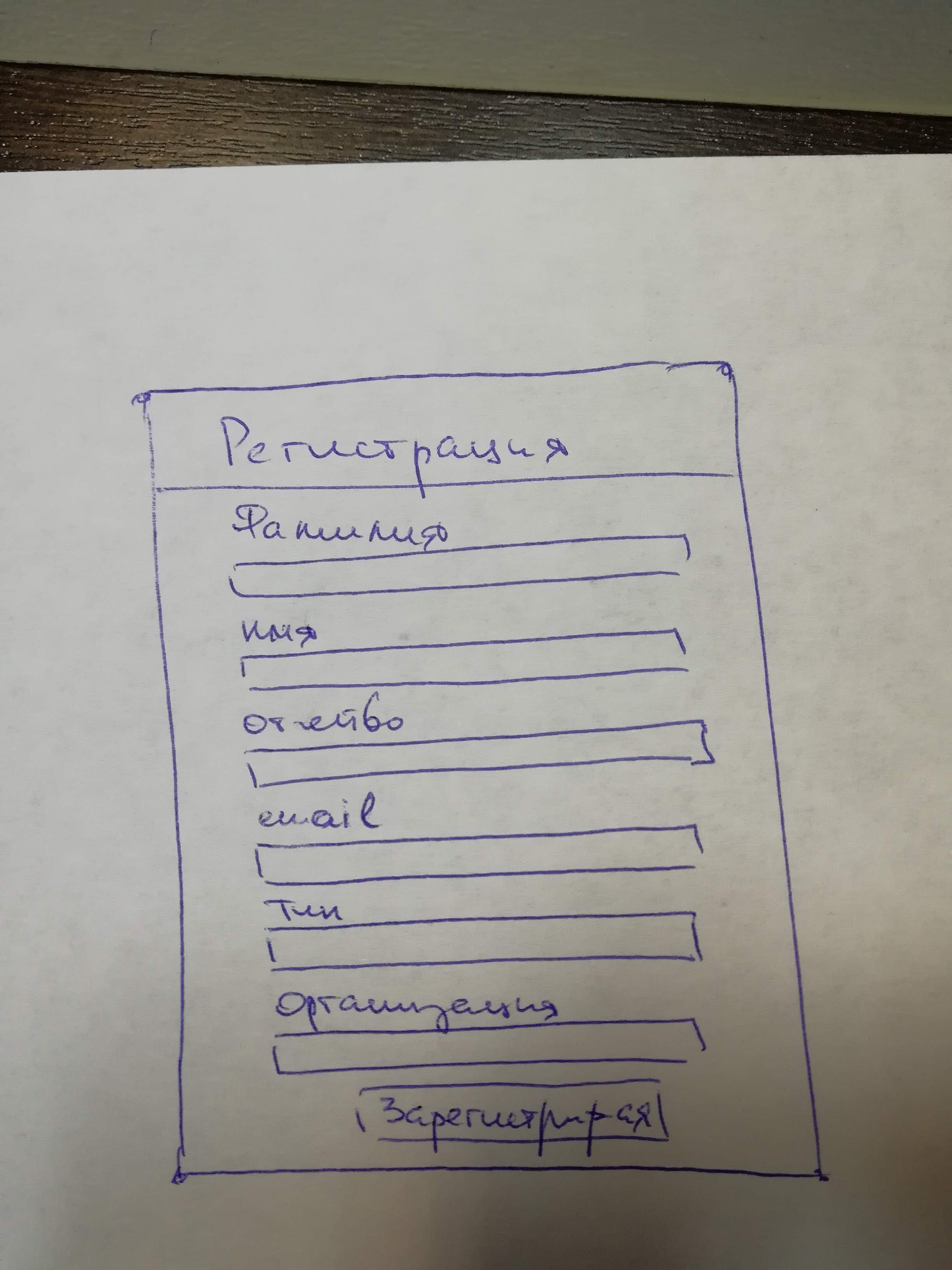
Тип учреждения, организация, должность, интерес по направлениям.

Поля формата JSON будут добавляться и меняться.

При регистрации пользователя ключевым полем для проверки является емейл. При вводе данных должно проверяться наличия уже зарегистрированного емейла, а также проверка вводимых данных по форматам.

После заполнения формы пользователю отправляется информационное письмо с просьбой подтвердить регистрацию. Подтверждение регистрации должно осуществляться сравнением кода в базе данных и кода, присланному в ссылке для подтверждения. Для пользователя ничего дополнительно вводить не требуется.

Текс письма должен быть вынесен в отдельный шаблон. Письмо отправляется в формате HTML с вложением файла электронного билета.

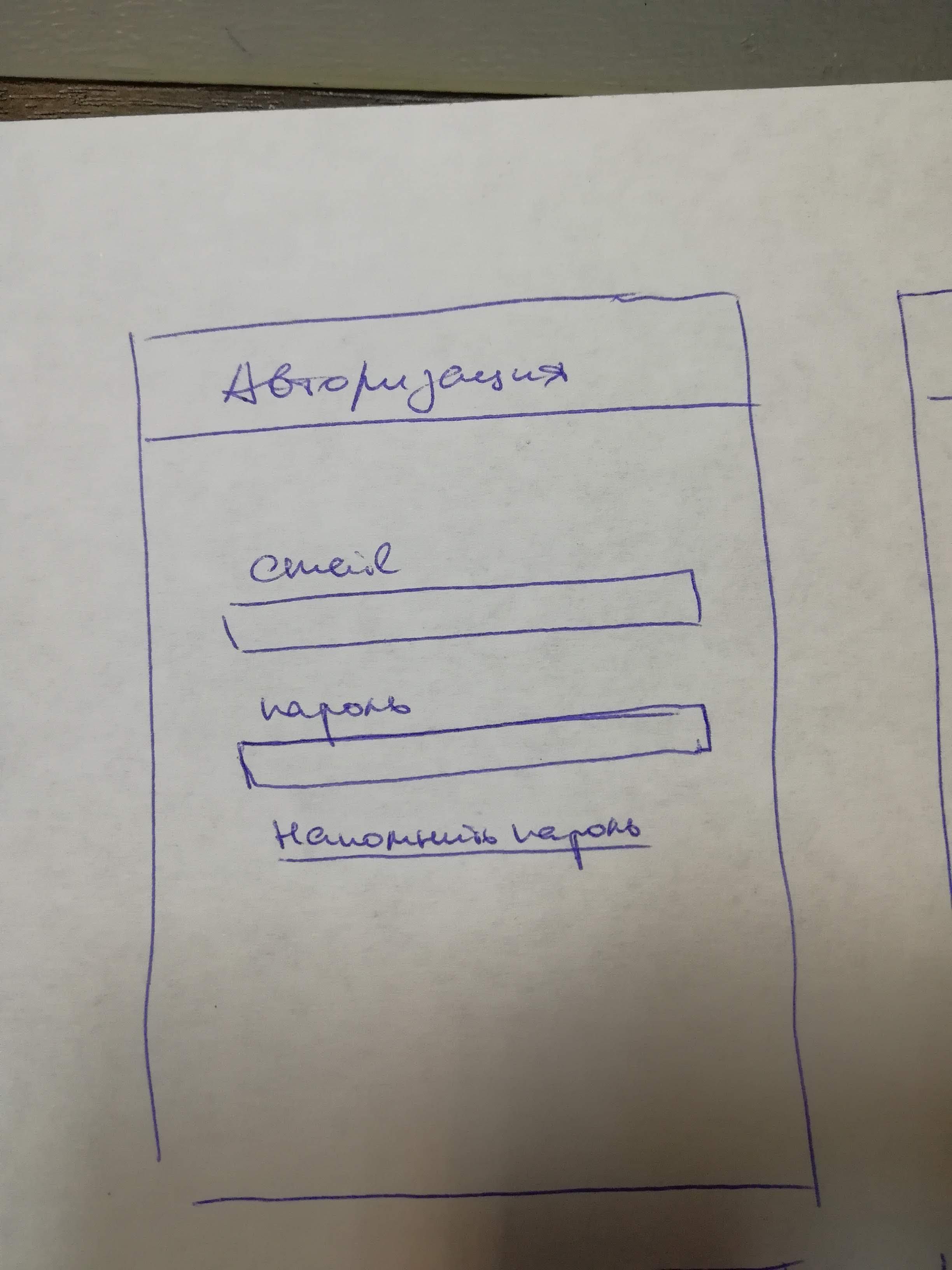


1. **Вход в личный кабинет.**

Вход в личный кабинет осуществляется по ссылке <http://servername.ru/my/> При загрузке страницы по указанному адресу необходимо проверять наличие авторизации. При её отсутствии пересылать запрос на страницу авторизации.

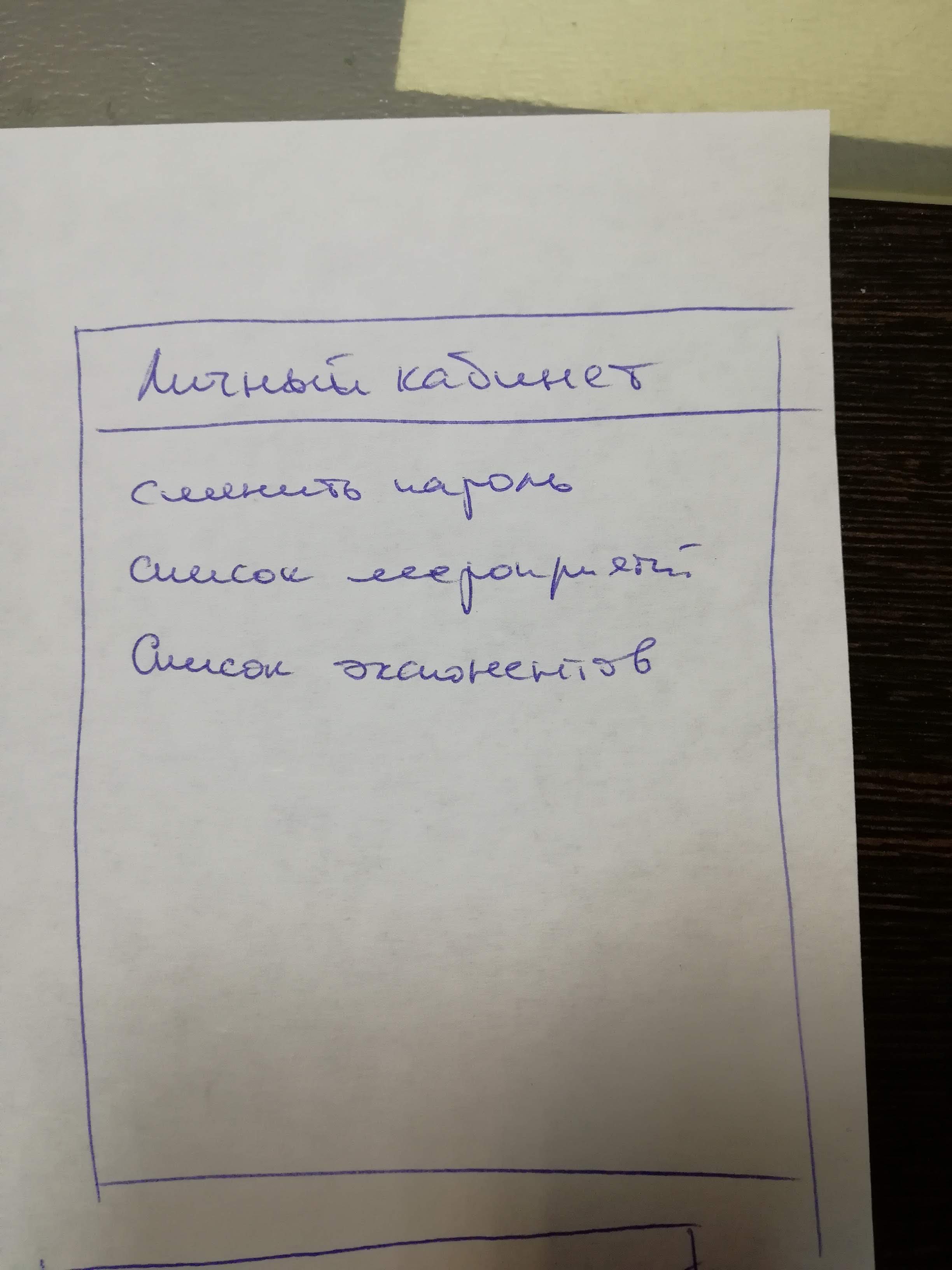
Допустимо использовать специальные коды, сгенерированные на основе данных пользователя для входа в личный кабинет без ввода паролей. Для примера:

<http://servername.ru/autologin/?s=888f50e45451a435e8d194914c4e163a&email=xxx@mail.ru>



1. **Разделы личного кабинета**

Вид главной страницы Лк (адрес http://servername.ru/my/)

****

* 1. Личные данные (адрес http://servername.ru/my/)

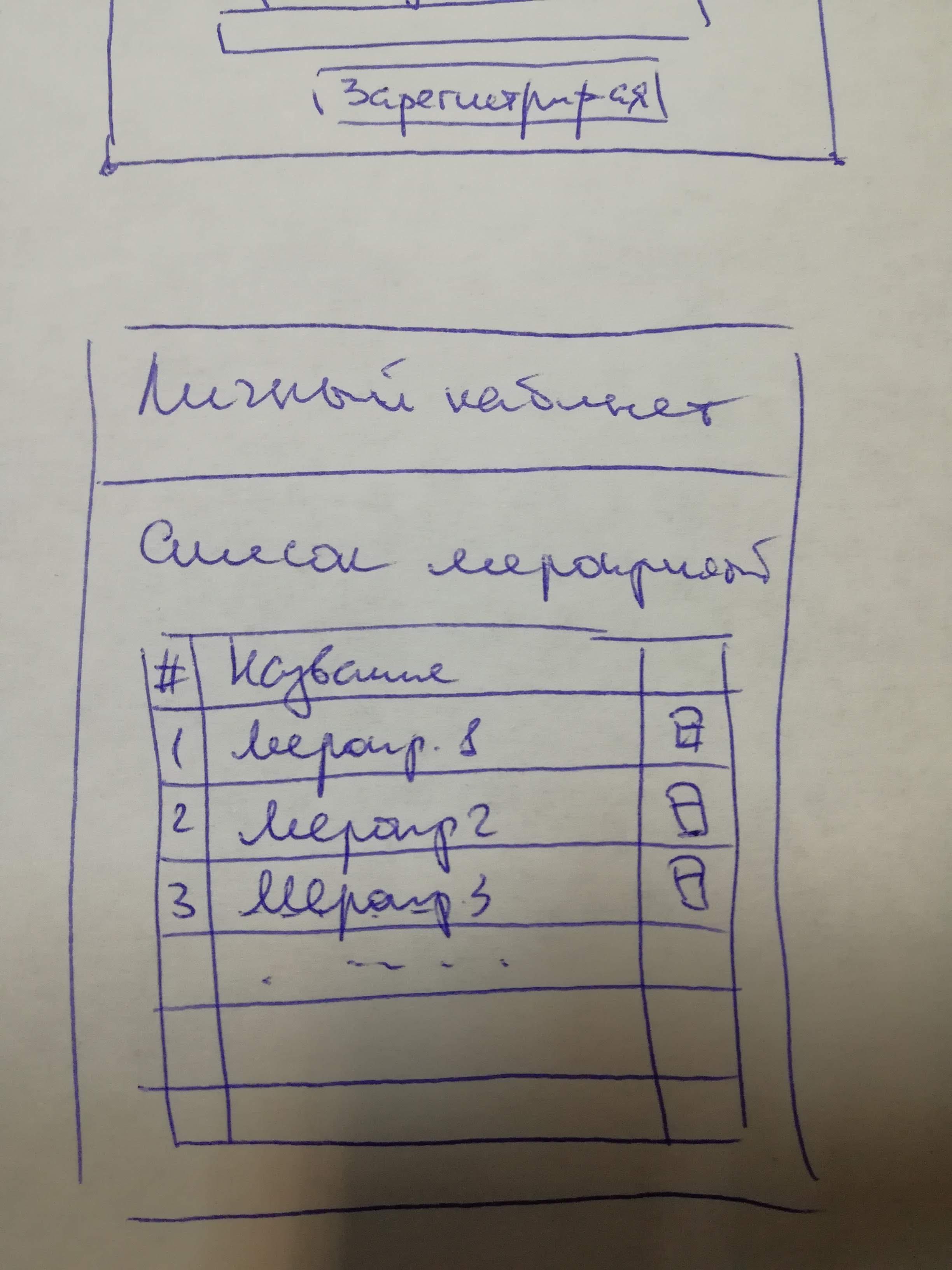
В данном разделе можно сменить пароль для аккаунта.

Стандартное решение с двумя полями ввода для пароля.

* 1. Мои мероприятия (адрес http://servername.ru/my/events)

Список мероприятий выбранные пользователем

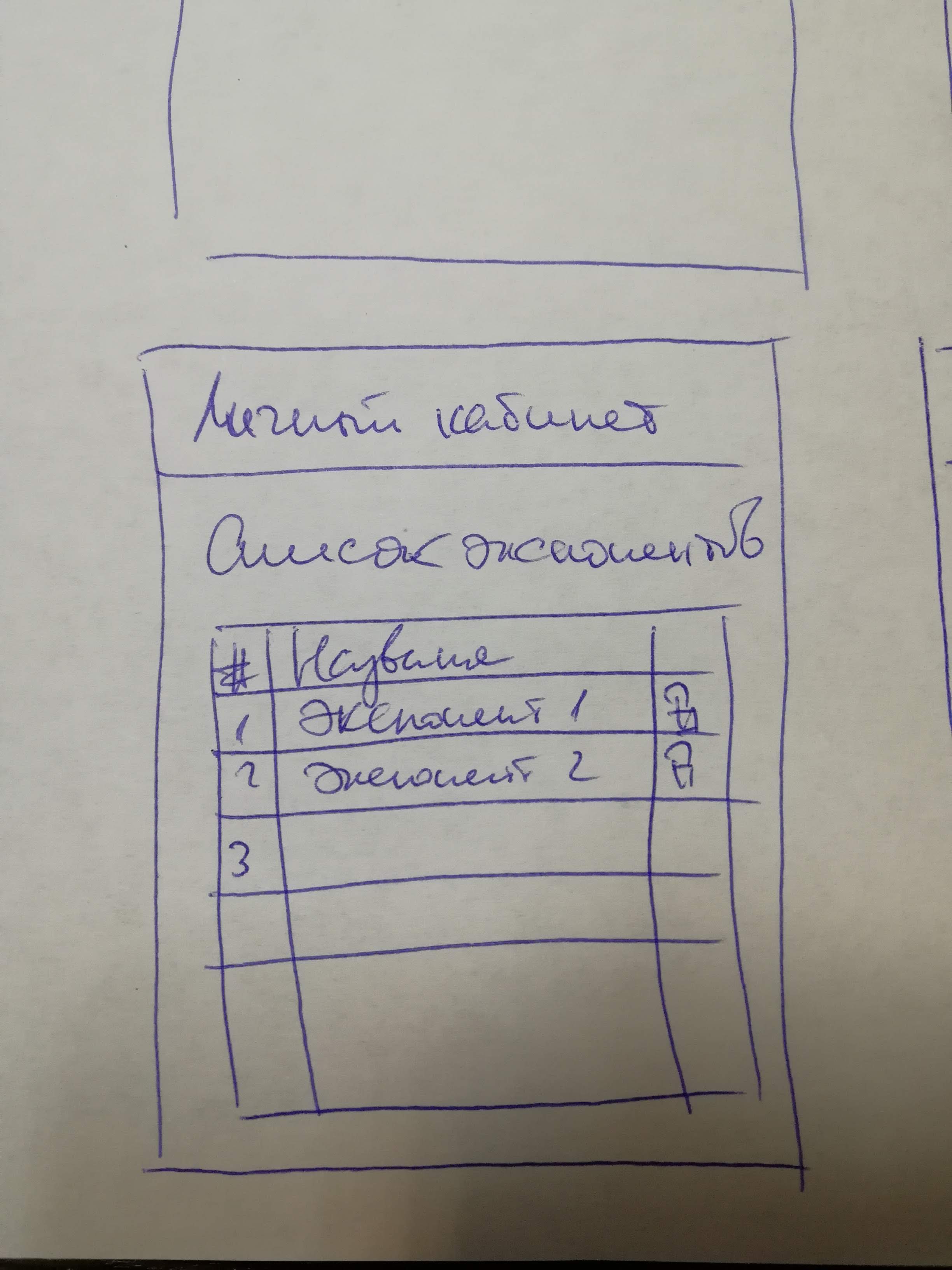
Возможность удалить мероприятие из списка



* 1. Мои экспоненты (адрес http://servername.ru/my/exponents)

Список экспонентов выбранные пользователем

Возможность удалить экспонента из списка

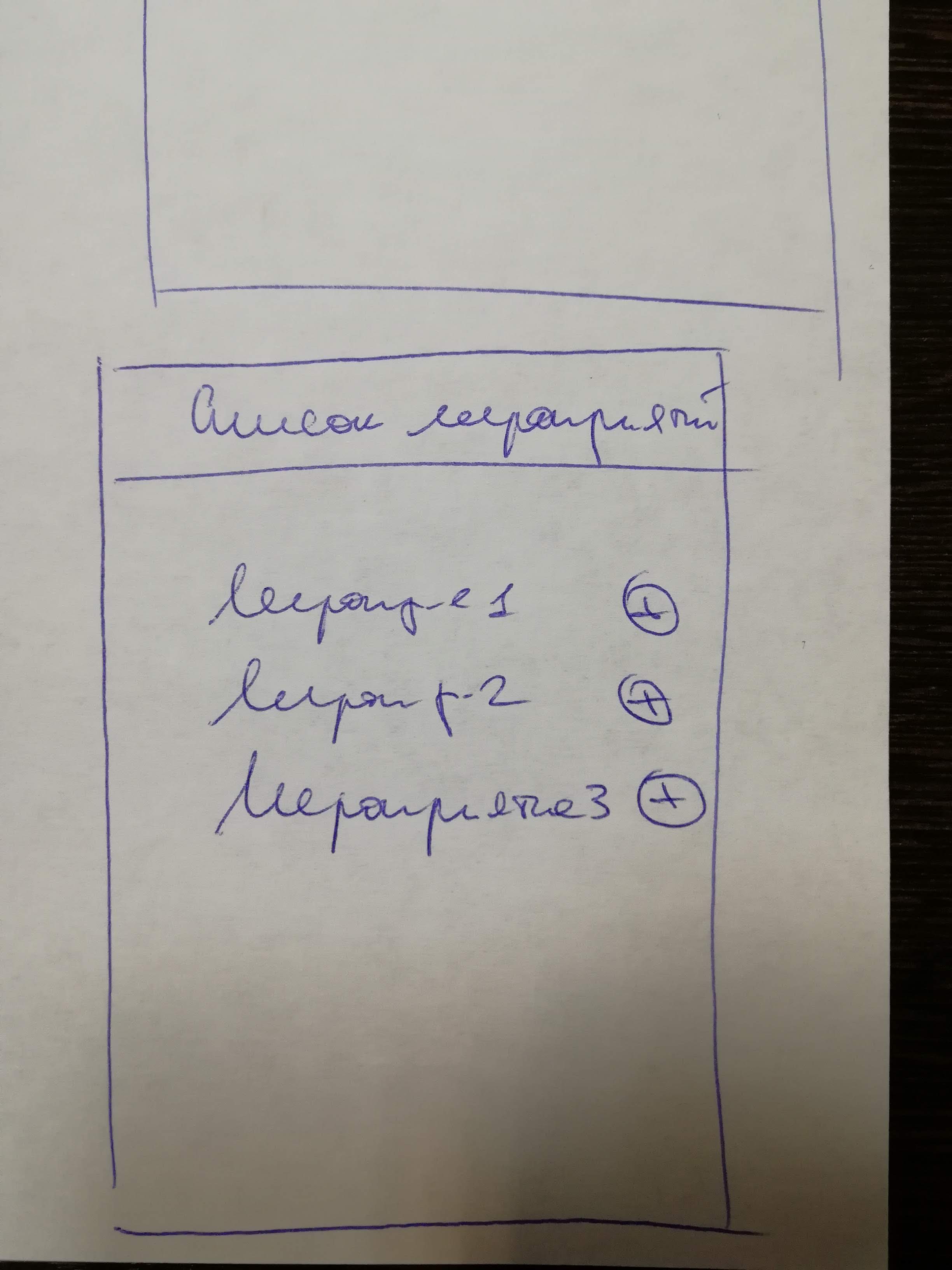


Удаление записей из списков должно осуществляться через ajax запросы без перезагрузки страницы.

1. **Добавление элементов в списки пользователя**

На отдельной странице мероприятий или экспонентов пользователь может добавить выбранное мероприятие в свой список. Добавление отдельного элемента осуществляется путем соответствующего ajax запроса.

Примерный вид страницы списка мерпроиятий с кнопкой добавления (адрес http://servername.ru/events/).



**Описание структуры БД.**

Таблица пользователей (users)

|  |  |  |
| --- | --- | --- |
| **Наименование** | **Тип** | **Комментарий** |
| Id | int | автоинкремент |
| surname | string | текстовое поле |
| name | string | текстовое поле |
| middlename | string | текстовое поле |
| email | string | текстовое поле |
| code | string | текстовое поле |
| data | text | поле для хранения набора данных JSON |
| datetime | datetime |  |

Таблица Мероприятий (events)

|  |  |  |
| --- | --- | --- |
| **Наименование** | **Тип** | **Комментарий** |
| Id | int | автоинкремент |
| name | string | наименование мероприятия |

Таблица Экспонентов (exponents)

|  |  |  |
| --- | --- | --- |
| **Наименование** | **Тип** | **Комментарий** |
| Id | int | автоинкремент |
| name | string | наименование экспонента |

Таблица связи Мероприятие Посетитель (event\_user)

|  |  |  |
| --- | --- | --- |
| **Наименование** | **Тип** | **Комментарий** |
| Id | int | автоинкремент |
| event\_id | int | id мероприятия |
| user\_id | int | id пользователя |

Таблица связи Посетитель Экспонент (exponent\_user)

|  |  |  |
| --- | --- | --- |
| **Наименование** | **Тип** | **Комментарий** |
| Id | int | автоинкремент |
| exponent\_id | int | id экспонента |
| user\_id | int | id пользователя |

Таблицы мероприятий и экспонентов можно заполнить тестовыми данными