# Министерство науки и высшего образования Российской Федерации

## Федеральное государственное бюджетное образовательное учреждение высшего образования

## «Новосибирский государственный технический университет»

NSTU_Logo_blue

## Кафедра теоретической и прикладной информатики

### Курсовой проект по дисциплине «Человеко-машинное взаимодействие»

**Разработка интерфейса системы сбора персональной информации абонента сотовой сети с использованием фреймворка AngularJS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| сигма градиент синий1 | Факультет: | ПМИ |  |  |
| Группа: | ПМИ-72 |  |  |
| Студент: | Антонов С.С. |  |  |
|  |  | Дата: |  |
| Преподаватель: | Волкова В.М. | Подпись: |  |

Новосибирск

2019

1. **Анализ предметной области**

Основная задача системы сотовой сети – управление тарифами и абонентами . Для этого нам необходимо реализовать следующие сценарии:

* Добавление абонента
* Удаление тарифа
* Поиск
* Просмотр абонентов
* Редактирование

Для разработки приложения воспользуемся HTML/CSS и средствами языка JavaScript, в частности фреймворком Angular.js.

1. **Use case диаграмма**
2. **Диаграммы деятельности**

**Добавление абонента**

Введите данные :

1)Имя  
2)Фамилия

3) Номер телефона

4) Выберите тариф

Поля пусты?

Ошибка абонента

Да

Нет

Добавить абонента

**Редактирование абонента**

Поля пусты? пусто?

Да

Нет

Отменить изменения

Сохранить изменения

Сохранить изменения?

Да

Нет

Выбрать абонента

Удалить?

Данные будут утеряны безвозвратно .Продолжить?

Да

Нет

Редактируем данные

Ошибка

**Поиск абонента**

Ввод данных

Данные пусты?

Поиск по названию

Заметка найдена?

Да

Нет

Переход к заметке

Да

Вывод пустого поля

Нет

**Удаление тарифа**

Выбрать записку

Удалить тариф

Тариф удален

Предупреждение

Нет

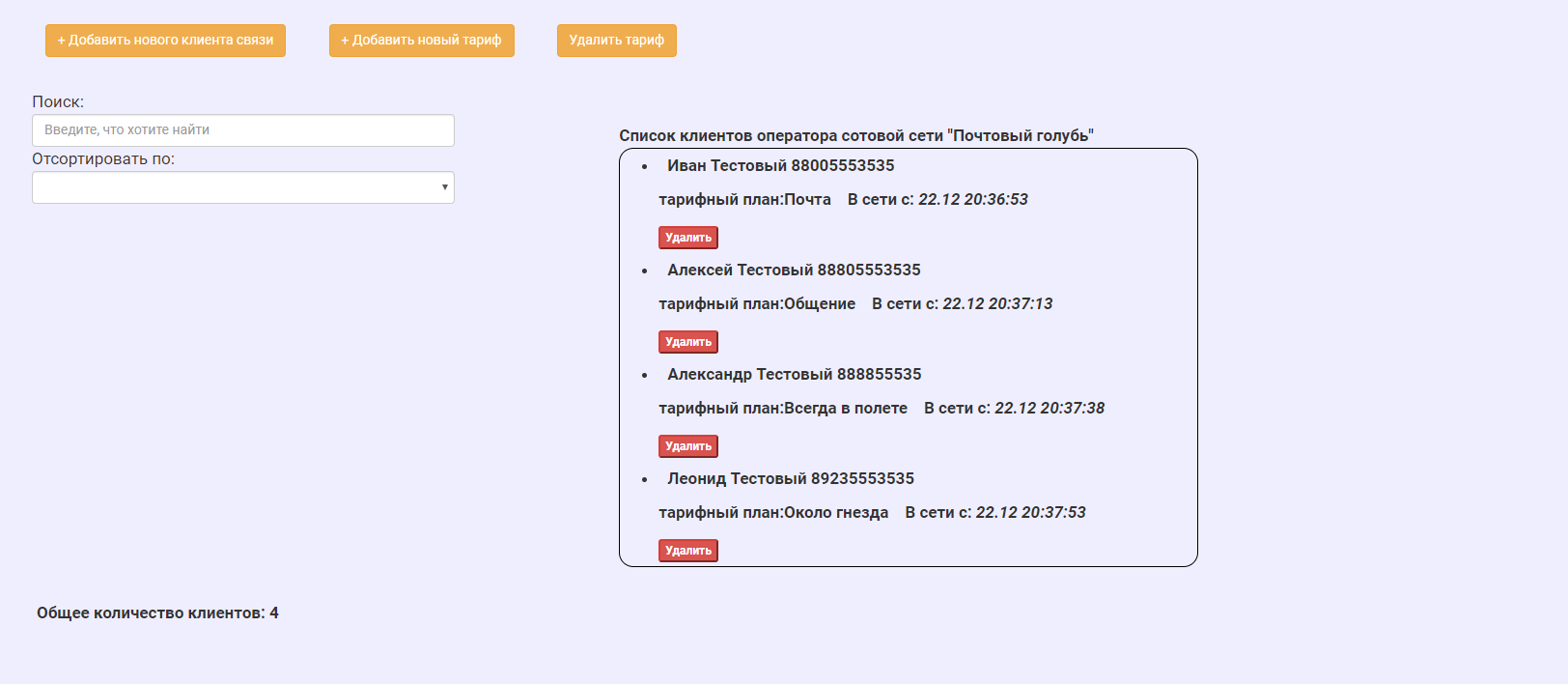
Да

**Просмотр абонентов**

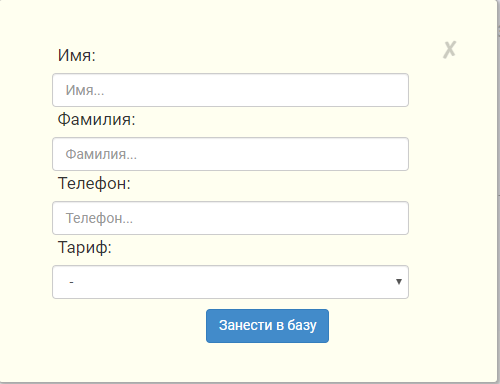
Сортировка абонентов

1. **Прототип интерфейса**

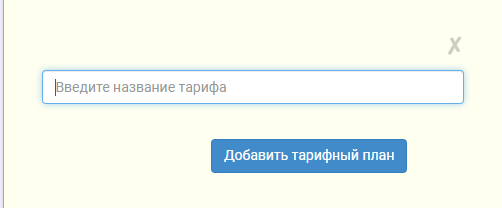
**Главная страница**



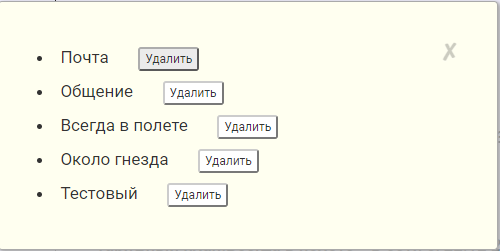
**Добавление абонента**



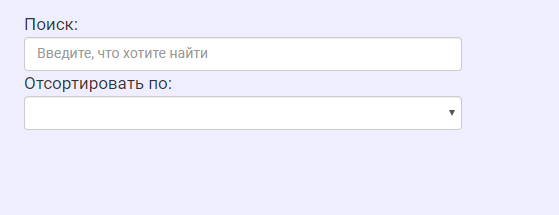
**Добавление тарифа**



**Удаление тарифа**

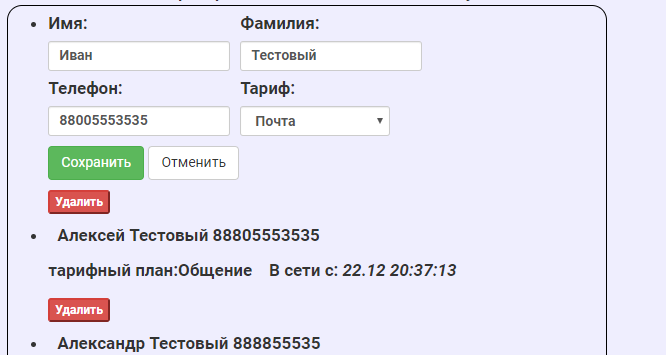


**Поиск**



Тут у нас есть возможность искать по имени или отсортировать по назанию тарифа или дате добавления абонента

**Редактирование**



1. **Использование средств фреймворка**

**AngularJS** — [JavaScript](https://ru.wikipedia.org/wiki/JavaScript" \o "JavaScript)-[фреймворк](https://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA) с [открытым исходным кодом](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5). Предназначен для разработки [одностраничных приложений](https://ru.wikipedia.org/wiki/%D0%9E%D0%B4%D0%BD%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%87%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5). Его цель — расширение браузерных приложений на основе [MVC](https://ru.wikipedia.org/wiki/Model-View-Controller)-шаблона, а также упрощение тестирования и разработки.

Фреймворк работает с [HTML](https://ru.wikipedia.org/wiki/HTML), содержащим дополнительные пользовательские атрибуты, которые описываются директивами, и связывает ввод или вывод области страницы с моделью, представляющей собой обычные переменные JavaScript. Значения этих переменных задаются вручную или извлекаются из статических или динамических JSON-данных

С помощью директив AngularJS можно создавать пользовательские HTML-теги и атрибуты, чтобы добавить поведение некоторым элементам