**Министерство транспорта Российской Федерации**

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«РУТ (МИИТ)»**

**Институт транспортной техники и систем управления**

**Кафедра «Управление и защита информации»**

Практическая работа

по дисциплине

**«Основы построения защищенных баз данных»**

**на тему  
«Школа»**

Выполнил: ст. гр. ТКИ-541  
Митин Г.А.  
Проверил: доцент кафедры

УиЗИ, к.т.н Васильева М.А.

Москва 2022

**Цель работы:**

Научится использовать структуру ОРМ на языке программирования Python.

**Постановка задачи:**

Написать на языке Python с использованием фреймворка Django базу данных Школа. Должна быть возможность создавать учебные классы, классных руководителей и учеников.

**Выполнение работы:**

**models.py**

from django.db import models  
  
class Teacher(models.Model):  
 name = models.CharField(max\_length=255, verbose\_name=**"Имя"**)  
 surname = models.CharField(max\_length=255, verbose\_name=**"Фамилия"**)  
 lastname = models.CharField(max\_length=255, verbose\_name=**"Отчество"**)  
  
 def \_\_str\_\_(self):  
 return **f"**{self.name} {self.surname} {self.lastname}**"**class Student(models.Model):  
 name = models.CharField(max\_length=255, verbose\_name=**"Имя"**)  
 surname = models.CharField(max\_length=255, verbose\_name=**"Фамилия"**)  
 lastname = models.CharField(max\_length=255, verbose\_name=**"Отчество"**)  
  
 def \_\_str\_\_(self):  
 return **f"**{self.name} {self.surname} {self.lastname}**"**class ClassName(models.Model):  
 name = models.CharField(max\_length=3, verbose\_name=**"Названия учебного класса"**, unique=True)  
  
  
 def \_\_str\_\_(self):  
 return self.name  
  
  
class ClassRoom(models.Model):  
 teacher = models.ForeignKey(Teacher, on\_delete=models.CASCADE)  
 student = models.ManyToManyField(Student, related\_name=**"prev"**, blank=True)  
 classname = models.ForeignKey(ClassName, on\_delete=models.CASCADE)  
  
 def \_\_str\_\_(self):  
 return **f"**{self.teacher} {self.student} {self.classname}**"**

**Описание классов:**

***Teacher***

|  |  |  |
| --- | --- | --- |
| *Название поля* | *Тип данных* | *Обязательное* |
| *name* | *CharField(255)* | *Да* |
| *surname* | *CharField(255)* | *Да* |
| *lastname* | *CharField(255)* | *Да* |

***Student***

|  |  |  |
| --- | --- | --- |
| *Название поля* | *Тип данных* | *Обязательное* |
| *name* | *CharField(255)* | *Да* |
| *surname* | *CharField(255)* | *Да* |
| *lastname* | *CharField(255)* | *Да* |

***ClassName***

|  |  |  |
| --- | --- | --- |
| *Название поля* | *Тип данных* | *Обязательное* |
| *name* | *CharField(3)* | *Да* |

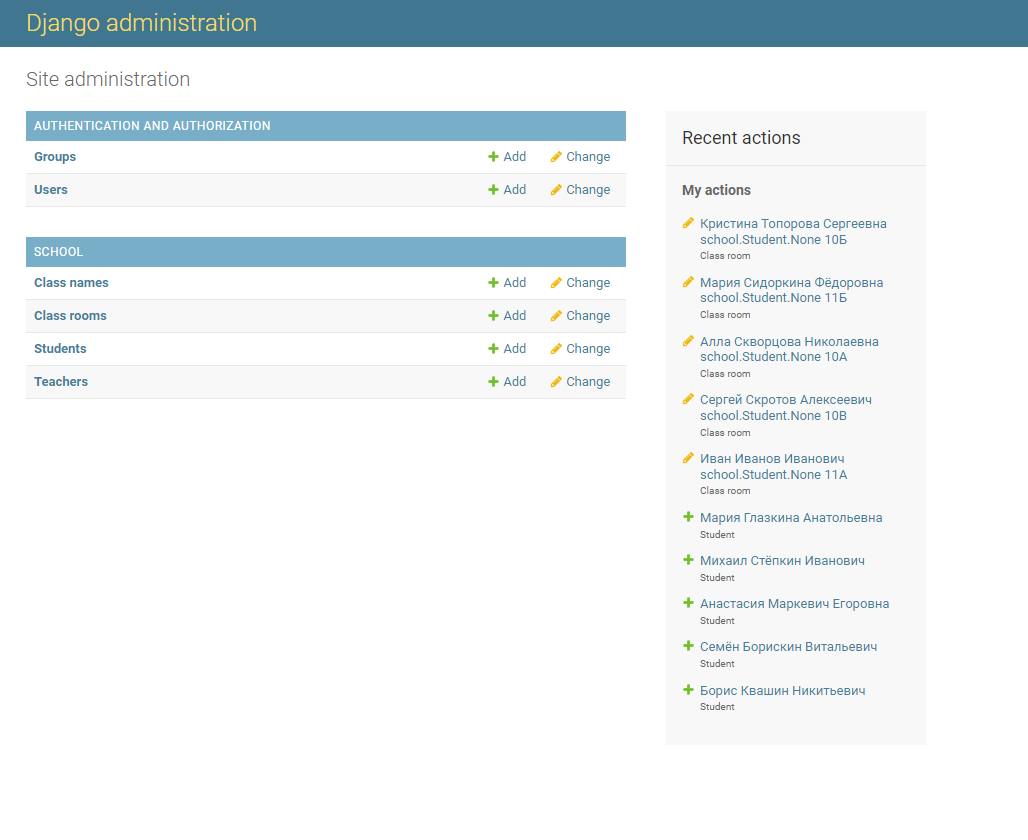
***ClassRoom***

|  |  |  |
| --- | --- | --- |
| *Название поля* | *Тип данных* | *Обязательное* |
| *teacher* | *ForeignKey(teacher)* | *Да* |
| *student* | *ForeignKey(student)* | *Да* |
| *classname* | *ForeignKey(classname)* | *Да* |

**Скриншоты выполнения работы**

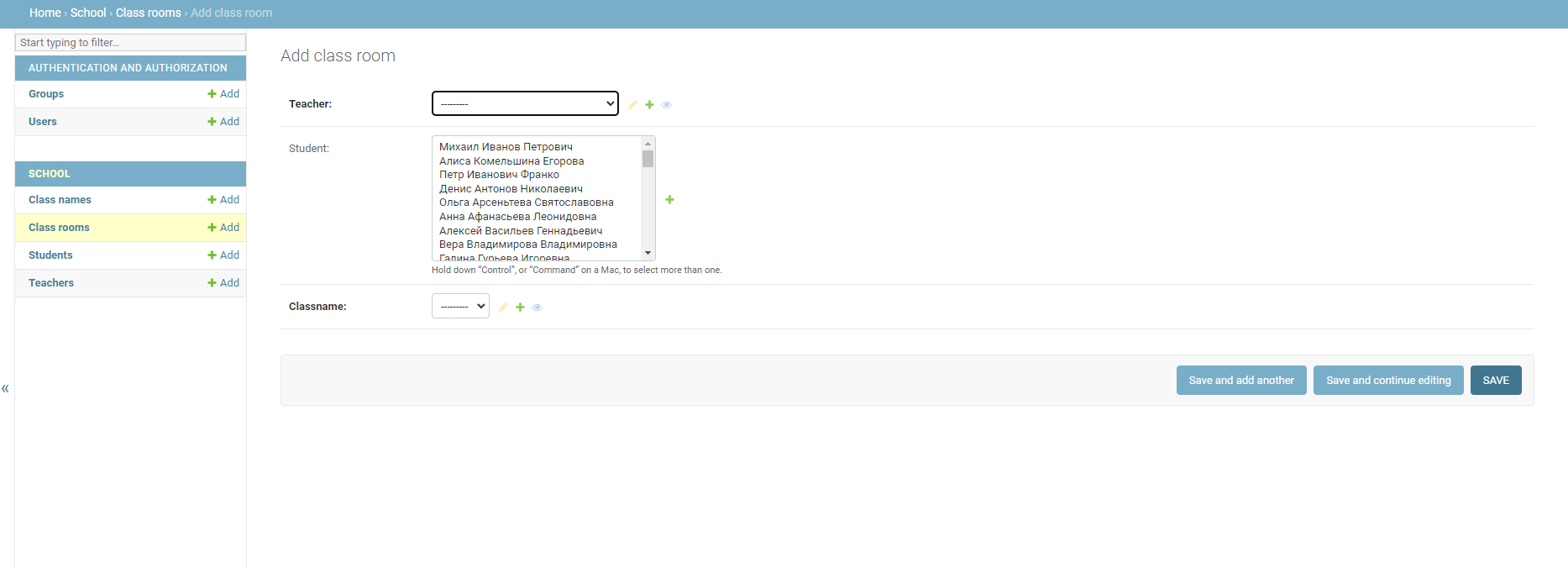
После создания классов в администрировании Django стало доступно изменение и добавления элементов в базу данных.

**Добавление данных**

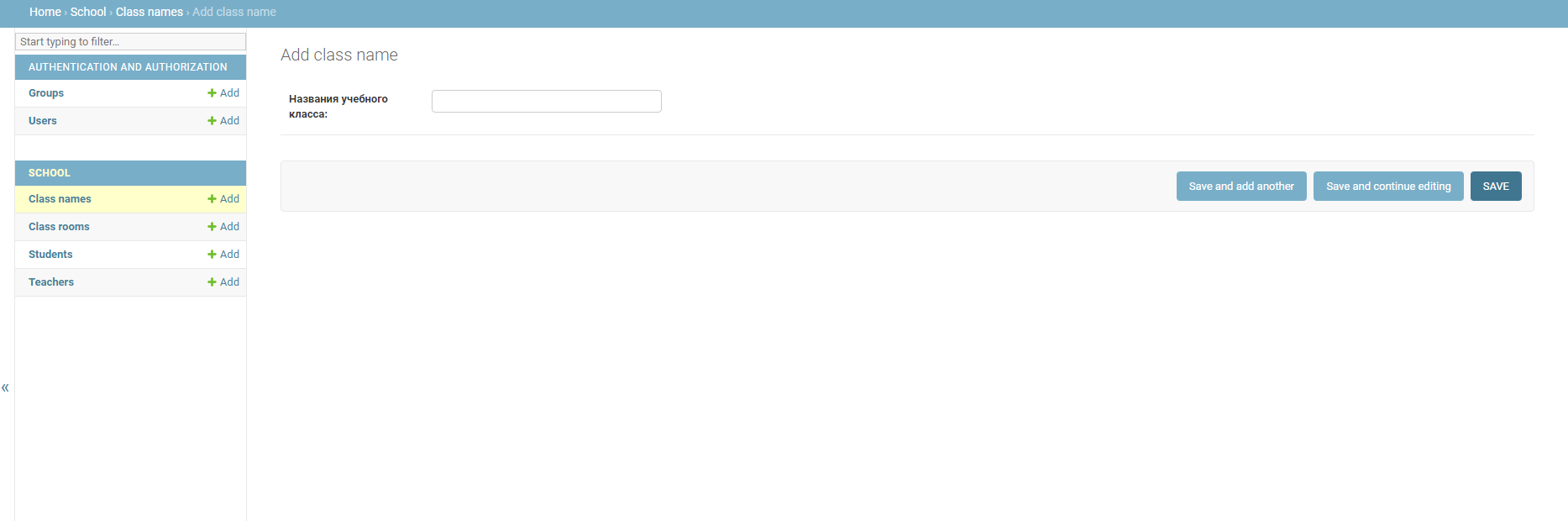


*Рисунок 1 страница администратора*

**Добавление данных учебного класса**

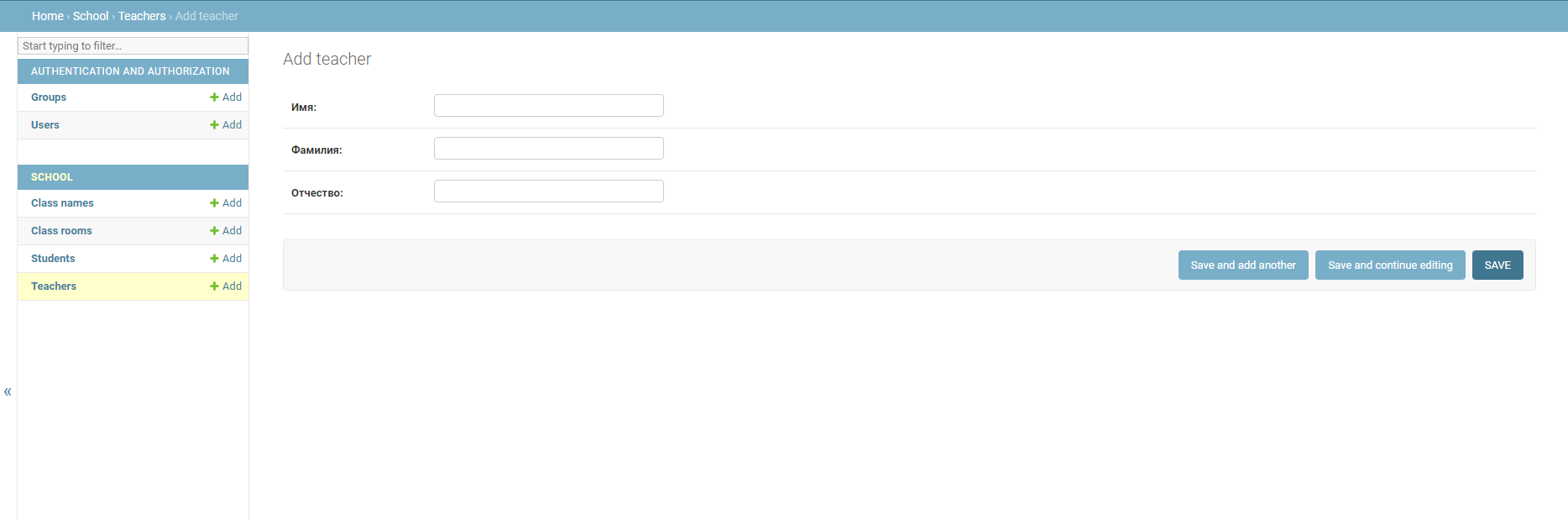
**

*Рисунок 2 Страница добавления данных учебного класса*

**

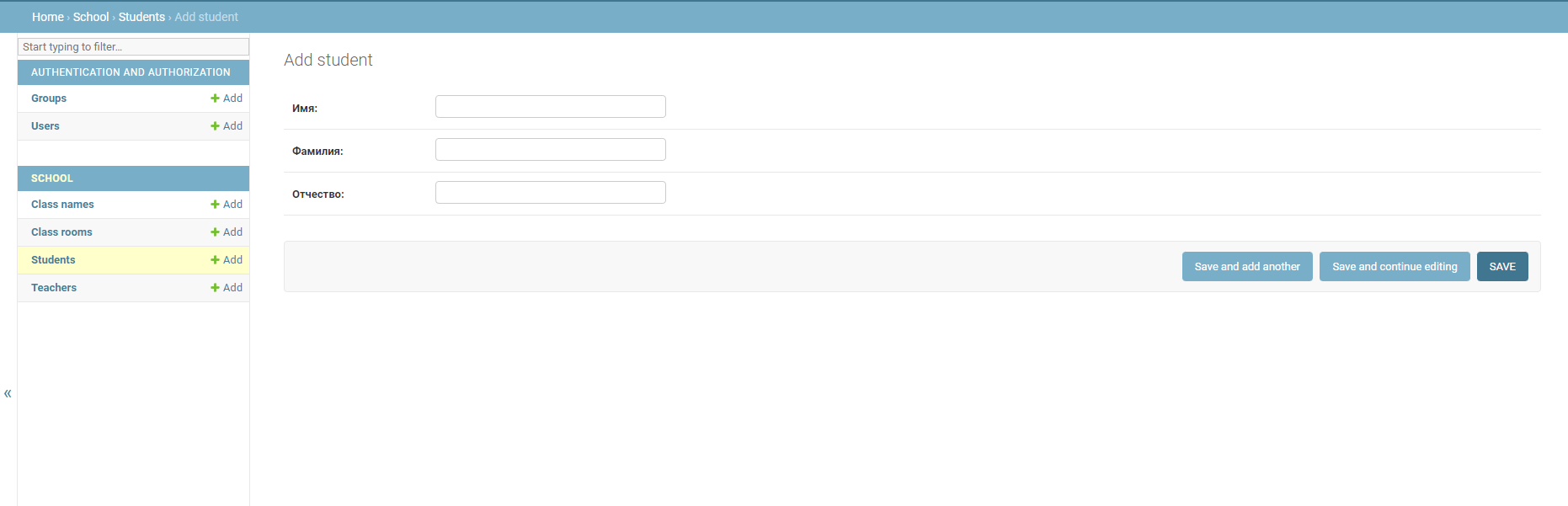
*Рисунок 3 страница добавления данных наименования класса*

**Добавление данных классного руководителя**

**

*Рисунок 4 страница добавления данных классного руководителя*

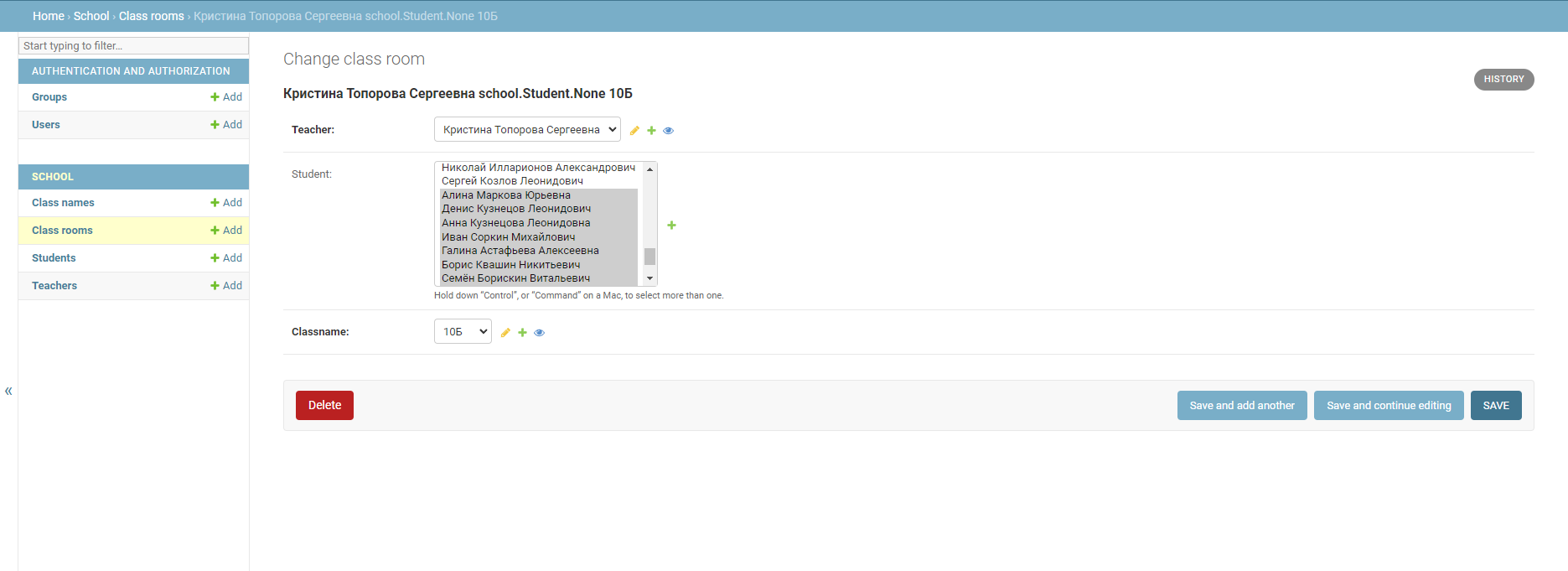
**Добавление данных учеников**

****

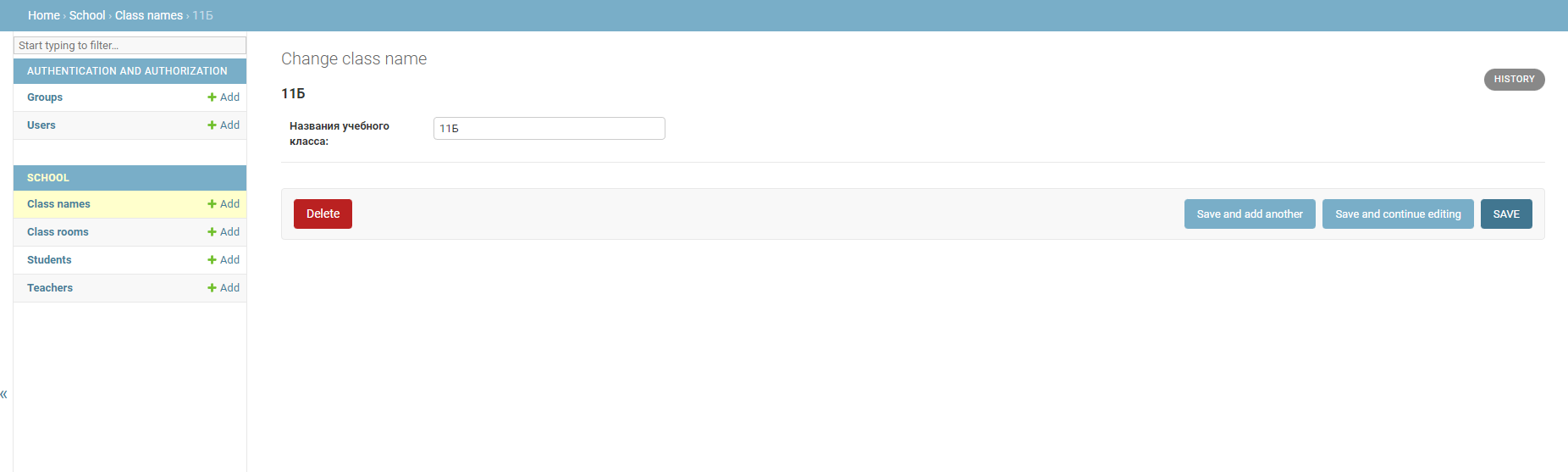
*Рисунок 5 страница добавления данных учеников*

**Изменение данных**

**Изменение данных учебного класса**

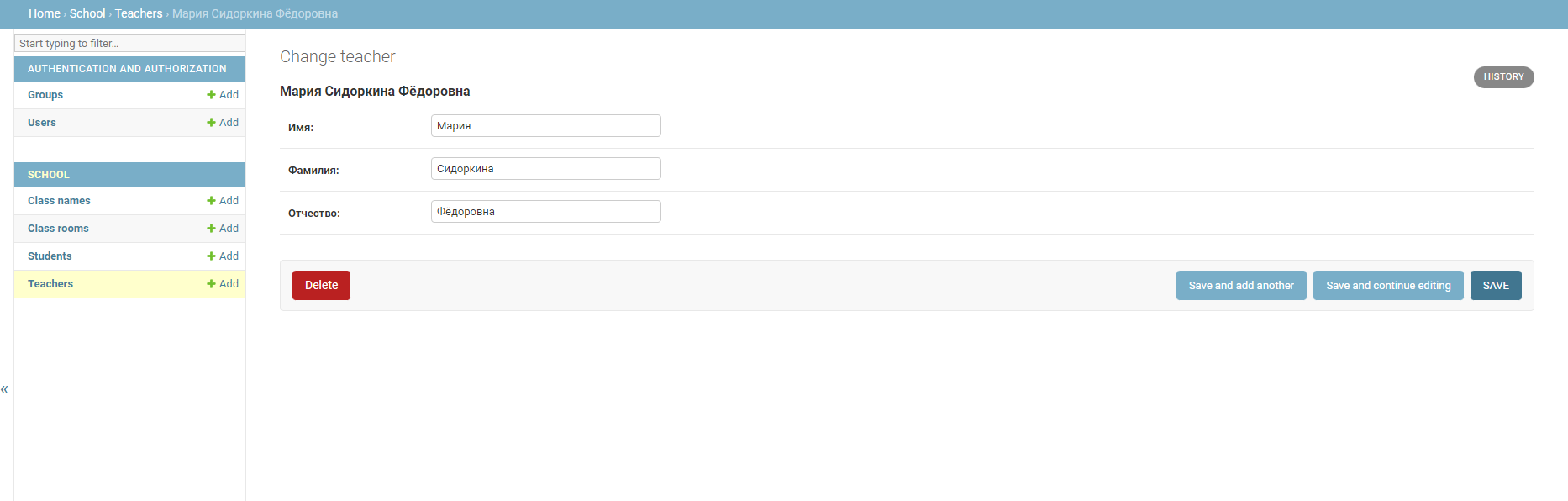
**

*Рисунок 6 Страница изменения данных учебного класса*

**

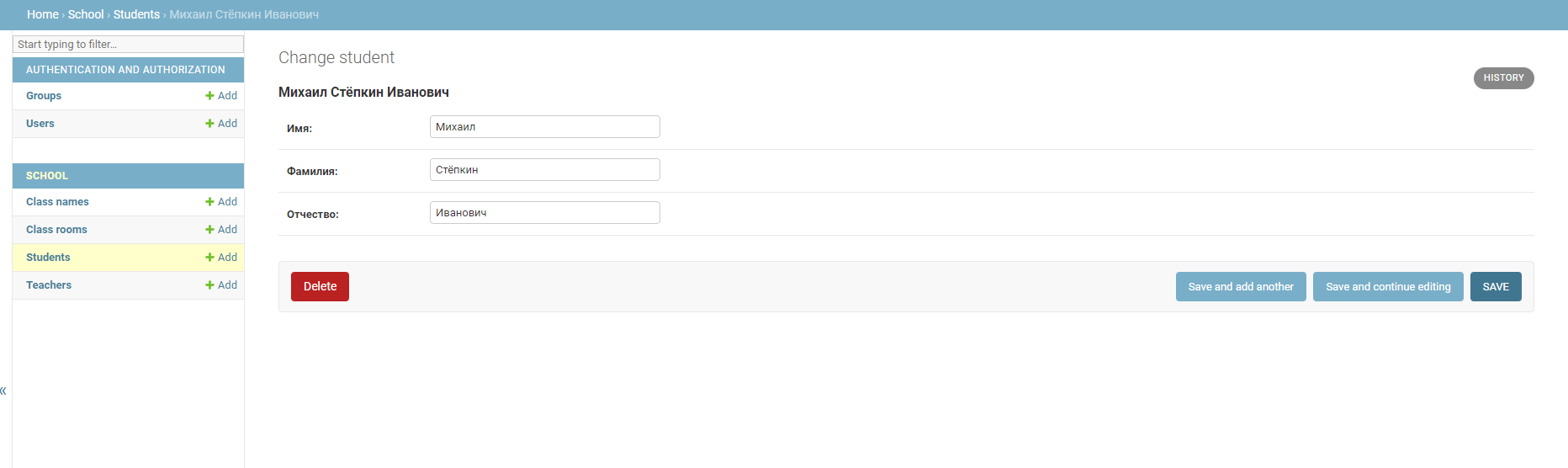
*Рисунок 7 страница изменения наименования класса*

**Изменение данных классного руководителя**

**

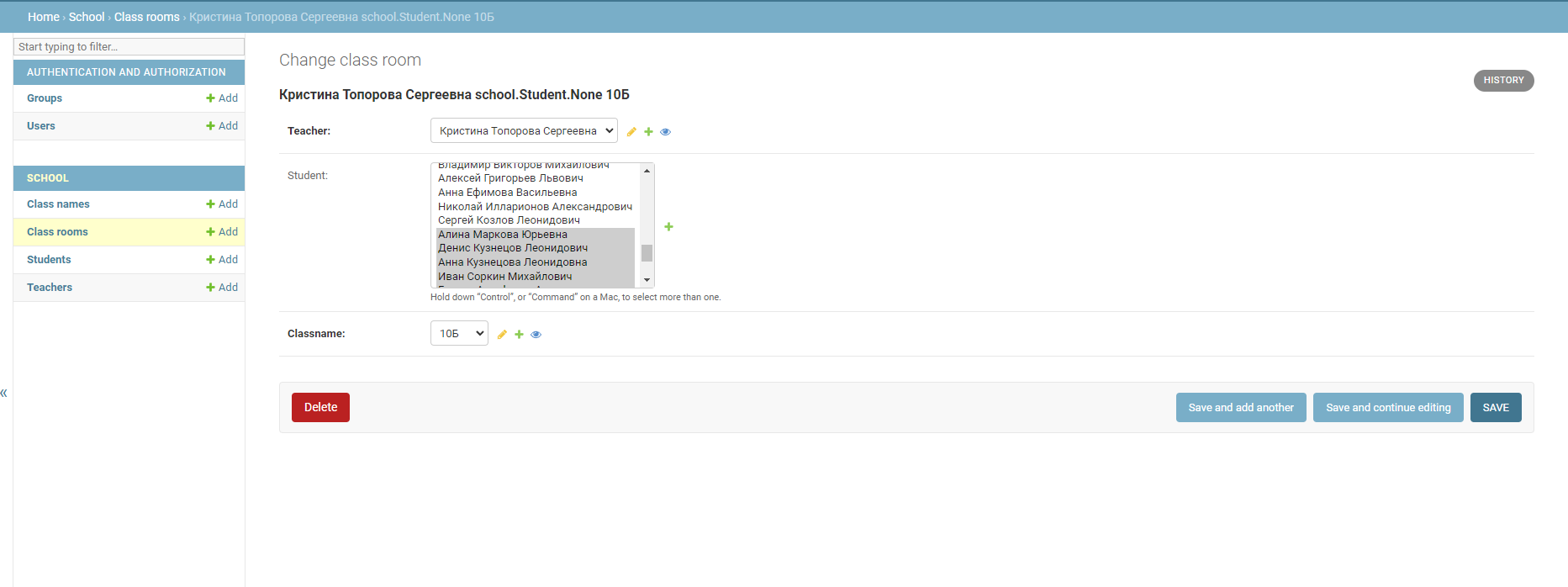
*Рисунок 8 страница изменения данных классного руководителя*

**Изменение данных учеников**

****

*Рисунок 9 страница изменения данных учеников*

**Данные об учебном классе**

****

*Рисунок 10 страница данных об учебном классе*

**Вывод:**

Была произведена работа по созданию базы данных с помощью технологии ОРМ Python Django.