**Оглавление**

[1. Аннотация 2](#_Toc532508415)

[2. Введение 3](#_Toc532508416)

[3. Анализ предметной области 3](#_Toc532508417)

[4. Функциональная модель предметной области 5](#_Toc532508418)

[4.1. Спецификация функциональной модели предметной области 5](#_Toc532508419)

[4.2. Описание модели DFD 5](#_Toc532508420)

[5. Инфологическая модель предметной области 7](#_Toc532508421)

[5.1. Спецификация инфологической модели 7](#_Toc532508422)

[6. Выбор инструментов разработки 9](#_Toc532508423)

[6.1 Выбор СУБД 9](#_Toc532508424)

[6.2. Выбор языка программирования 9](#_Toc532508425)

[6.3. Паттерн MVC 9](#_Toc532508426)

[7. Пример работы и экранные формы интерфейса. 11](#_Toc532508427)

[7.1 Инструкция по запуску 11](#_Toc532508428)

[*7.2 Экранные формы* 12](#_Toc532508429)

[*7.2.1 Экранные формы пользователя студент* 12](#_Toc532508430)

[*7.2.2 Экранные формы пользователя преподаватель* 16](#_Toc532508431)

[7.3 Добавление новой информации 18](#_Toc532508432)

[8. Руководство пользователя 18](#_Toc532508433)

[9. Заключение 19](#_Toc532508434)

# 1. Аннотация

База данных «Система учета успеваемости студентов» может быть полезна студентам и преподавателям.

С использованием продукта можно отслеживать расписание занятий, оценки, выставленные за занятия, рейтинг студентов. Так же может быть получена информация о тех или иных предметах, преподавателях.

Программный продукт представляет собой клиент-серверное веб приложение написанное на языке Python, в качестве хранилища данных выбрана СУБД SQLite.

# 2. Введение

При выполнении курсовой работы были поставлены следующие цели:

- получение навыков проектирования схемы хранения данных

- получение навыков создания приложений к базам данных на Python

- получение навыков грамотного оформления документации: описание предметной области; инфологической, даталогической моделей; структурной схемы системы, составления DFD (Data Flow Diagram)

В результате выполнения курсовой работы должна быть создана АИС «Система учета успеваемости студентов», которая позволит автоматизировать работу учета успеваемости студентов.

Данный программный продукт позволит студентам получить информацию о занятиях, преподавателях, расписании, их оценках, несданных экзаменах. Информация проведении занятия так же выдается студенту или преподавателю при просмотре информации о занятии в расписании.

Преподаватели при помощи данного продукта смогут вести учет успеваемости студентов, добавлять новые занятия в расписание, ставить оценки за них.

Данная автоматизированная система позволит значительно упростить и систематизировать работу с учетом успеваемости студентов.

Программный продукт представляет собой базу данных, под управлением СУБД SQLite, а также веб-приложение на Python.

# 3. Анализ предметной области

Введение в эксплуатацию данной АИС позволит автоматизировать сбор и хранение информации об оценках студентов, расписании и занятиях, предметах, преподавателях.

При выставлении новых оценок студентам, их можно с легкостью занести в таблицу при помощи специальной формы. Если оценка была выставлена неверно, ее можно легко удалить. Предусмотрена функция быстрого удаления и добавления занятий в расписание с помощью специальных форм, что предоставит возможность не только упростить, но и ускорить работу с обработкой информации.

Система предоставляет возможность добавлять новых преподавателей, студентов, предметы. При добавлении новых преподавателей есть возможность так же указать предмет, который ведет данный преподаватель, и наоборот.

Таким образом система ускоряет и упрощает работу ведения учета успеваемости студентов.

*Требования заказчиков.*

Система предназначена для пользования двумя основными видами пользователей:

1. Преподаватель (администратор системы). Этим пользователям необходимо, чтобы система удовлетворяла следующим требованиям:

- возможность добавления новых занятий

-возможность изменения информации о занятии

- возможность добавления новых студентов

- возможность добавления новых предметов

- возможность изменения информации о преподавателе

- возможность изменения информации о предмете

2. Студент. Этим пользователям необходимо, чтобы система удовлетворяла следующим требованиям:

- возможность просмотра расписания

- возможность просмотра информации занятии в расписании

- возможность просмотра информации о предмете

- возможность просмотра информации о преподавателе

- возможность просмотра информации о текущей успеваемости

- возможность просмотра информации о средней оценке

# 

# 4. Функциональная модель предметной области

## 4.1. Спецификация функциональной модели предметной области

*1. Редактирование данных*

1.1. Редактирование данных по Студентам.

1.2. Редактирование данных по Преподавателям.

1.3. Редактирование данных по Предметам.

1.4. Редактирование данных по Занятиям.

1.5. Редактирование данных по Оценкам.

*2. Экранные формы*

2.1. Форма для ввода и редактирования данных по Студентам

2.2. Форма для ввода и редактирования данных по Преподавателям

2.3 Форма для ввода и редактирования данных по Предметам

2.4. Форма для ввода и редактирования данных по Занятиям

2.5. Форма для ввода и редактирования данных по Оценкам

## 4.2. Описание модели DFD

*Объекты:*

Студент, преподаватель – имеет возможность просмотра информации, касающейся расписания, занятиях в расписании, преподавателей, предметах, оценках.

Администратор – управляет работой всей системы. Занимается пополнением и изменением информационной части базы данных.

*Функции:*

Для студента или преподавателя:

Просмотр расписания

Просмотр занятий

Просмотр информации о преподавателях и предметах

Просмотр оценок

Для администратора:

Добавление и изменение информации расписания;

Добавление и изменение информации занятий;

Добавление и изменение информации предметов;

Добавление и изменение информации преподавателей;

Добавление и удаление студентов;

*Хранимые данные:*

Предметы;

Занятия;

Преподаватели;

Оценки;

Студенты;

*Потоки данных:*

Запрос информации о предмете – данные о предмете;

Запрос информации о преподавателе– данные о преподавателе

Запрос информации о расписании – данные о расписании

Запрос информации о студентах – данные о студентах

Запрос информации о занятии – данные о занятии

Добавление\Удаление студента – обновленный список студентов

Добавление\Изменение предмета – измененный список предметов

Добавление\Изменение преподавателя – измененный список преподавателей

Добавление\Изменение занятия – измененное расписание

# 5. Инфологическая модель предметной области

## 5.1. Спецификация инфологической модели

*1. Атрибуты*

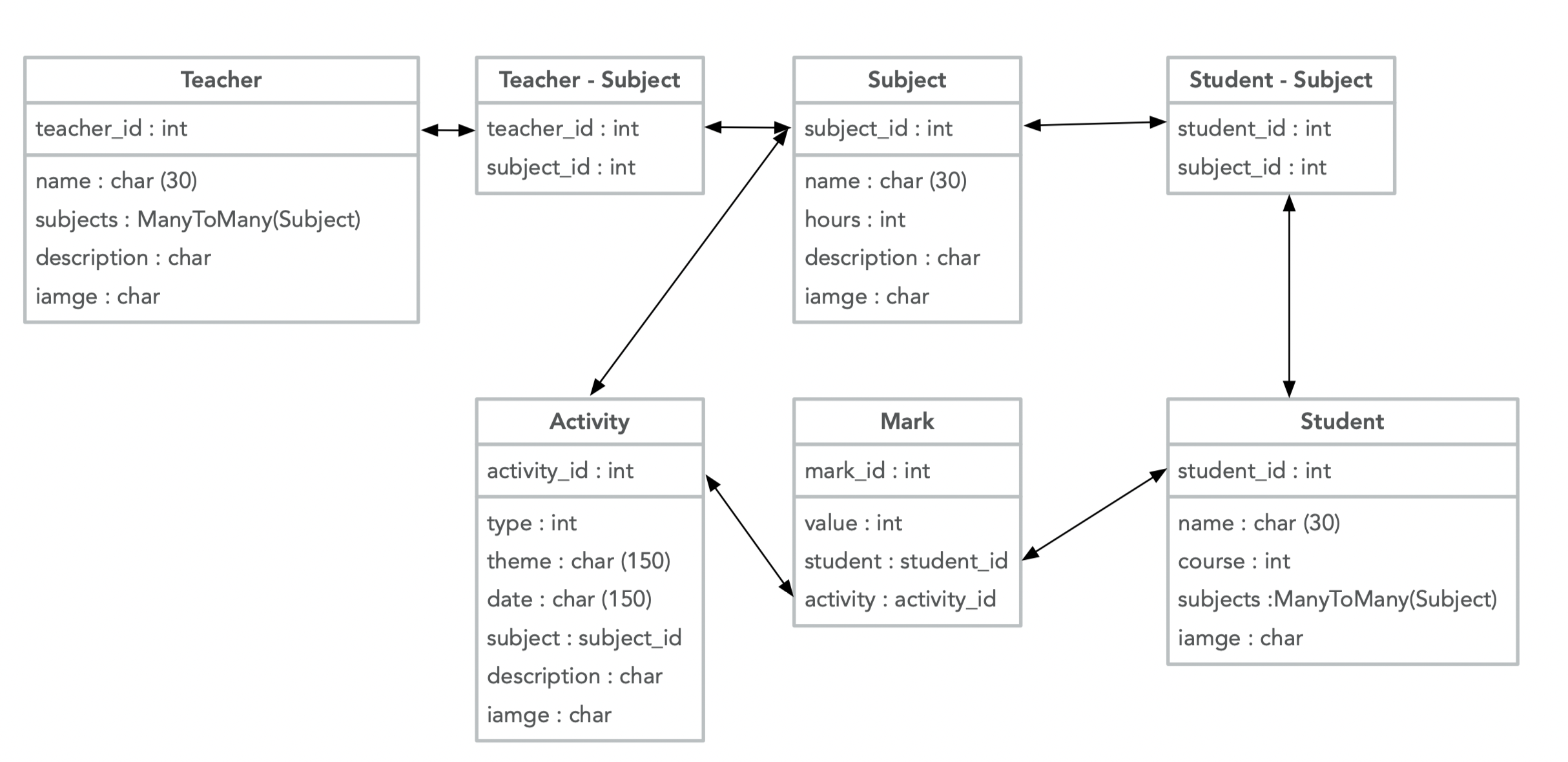
* 1. Код предмета
  2. Название предмета
  3. Количество часов
  4. Описание предмета
  5. Изображение предмета
  6. Код преподавателя
  7. Имя преподавателя
  8. Предметы преподавателя
  9. Описание преподавателя
  10. Фото преподавателя
  11. Код занятия
  12. Тип занятия
  13. Тема занятия
  14. Дата занятия
  15. Предмет занятия
  16. Преподаватель занятия
  17. Аудитория занятия
  18. Код студента
  19. Имя студента
  20. Курс студента
  21. Предметы студента
  22. Код оценки
  23. Значение оценки
  24. Студент оценки
  25. Занятие оценки

*2. Сущности*

* 1. Предмет(Код Предмета, Название Предмета, Количество часов, Описание Предмета, Изображение Предмета)
  2. Преподаватель(Код Преподавателя, Имя Преподавателя, Предметы Преподавателя, Описание Преподавателя, Фото Преподавателя)
  3. Занятие(Код занятия, Тип занятия, Тема Занятия, Дата Занятия, Предмет Занятия, Преподаватель Занятия, Аудитория Занятия)
  4. Студент(Код студенты Имя Студента, Курс Студента, Предметы Студента)
  5. Оценка(Код оценки, Значение Оценки, Студент Оценки, Занятие Оценки)

*3. Связи между сущностями*.

1. Предмет-Преподаватель, тип М:М
2. От Занятия к Предмету, тип 1:1
3. От Занятия к Преподавателю, тип 1:1
4. От Студента к Предмету, тип М:M
5. От Оценке к Студенту, тип 1:1



*Рис. 1. Схема инфологической модели данных.*

Видно, что все атрибуты всех сущностей атомарные (то есть неделимы) и не содержат повторяющихся групп. Следовательно, модель находится в первой нормальной форме.

Первичный ключ функционально и полно определяет все атрибуты, т.е. любой из атрибутов полностью зависит от первичного ключа, во всех сущностях предметной области. Следовательно, инфологическая модель нормализована ко второй нормальной форме.

Для всех сущностей все атрибуты зависят от первичного ключа и не зависят друг от друга. Таким образом, учитывая, что модель предметной области уже находится во второй нормальной форме, она нормализована и к третьей нормальной форме.

После проведенных преобразований видно, что все атрибуты зависят только от первичного ключа и отсутствуют многозначные зависимости, т.е. инфо­логическая модель системы находится в четвертой нормальной форме.

# 6. Выбор инструментов разработки

## 6.1 Выбор СУБД

Для реализации базы данных использована СУБД SQLite. Она отвечает всем необходимым требованиям для реализации, сущностей, связей между ними, запросов, реализации отчетов и удобных для представления пользователю форм. Также для базы данных сделано приложение на Python и сайт, что облегчает конечную визуализацию итоговой базы данных пользователю в виде единого независимого файла.

## 6.2. Выбор языка программирования

В качестве языка для разработки был выбран Python по следующим причинам:

* Оптимальное соотношение скорость разработки / сложность разработки.
* Кроссплатформенность
* Большое количество готовых решений в виде библиотек.
* Большое кол-во документации в интернете.

При разработке серверов web приложений самым популярным архитектурным паттерном является MVC. Django – самый популярный и широко используемый фреймворк на языке Python, реализующий паттерн MVC. Фреймворк — это набор компонентов, которые помогают разрабатывать веб-сайты быстро и просто.

Когда на сервер приходит запрос, он переадресуется Django, который пытается сообразить, что же конкретно от него просят. Для начала он берет адрес веб-страницы и пробует понять — что же нужно сделать. Эту часть процесса в Django выполняет ***urlresolver*** (адрес веб-сайта называется URL — Uniform Resource Locator — Единый указатель ресурсов, так что название urlresolver, resolver == определитель, имеет определенный смысл). Django сверяет шаблоны адресов сверху вниз и, если что-то совпадает, он переправляет запрос соответствующей функции (которая называется view).

## 6.3. Паттерн MVC

MVC - логика доступа к данным, бизнес-логика и логика отображения — составляют концепцию, которую называют шаблоном Модель-Представление-Управление (Model-View-Controller, MVC) архитектуры программного обеспечения. В этой концепции термин «Модель» относится к логике доступа к данным; термин «Представление» относится к той части системы, которая определяет, что показать и как; а термин «Управление» относится к той части системы, которая определяет какое представление надо использовать, в зависимости от пользовательского ввода, по необходимости получая доступ к модели.

Django следует модели MVC достаточно близко, т.е., может быть назван MVC совместимой средой разработки. Вот как M, V и C используются в Django:

- *M,* доступ к данным, обрабатывается слоем работы с базой данных, файл *models.py*.

- *V,* эта часть, которая определяет какие данные получать и как их отображать, обрабатывается представлениями и шаблонами, файлы *views.py* и *templates*.

- *C*, эта часть, которая выбирает представление в зависимости от пользовательского ввода, обрабатывается самой средой разработки, следуя созданной вами схемой URL, и вызывает соответствующую функцию Python для указанного URL, файл *urls.py.*

# 7. Пример работы и экранные формы интерфейса.

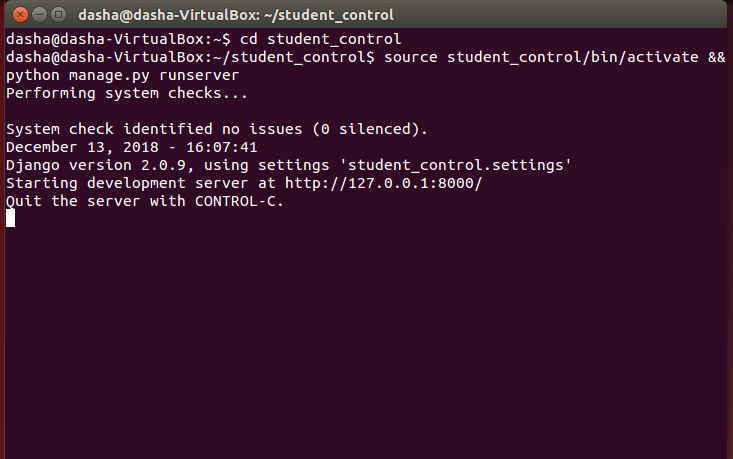
## 7.1 Инструкция по запуску

В терминале Ubuntu последовательно ввести команды:

1. *cd student\_control –* переход в папку проекта

*2. source student\_control/bin/activate* – активация виртуального окружения Python

*3. python manage.py runserver –* запуск HTTP сервера приложения на порту 8000

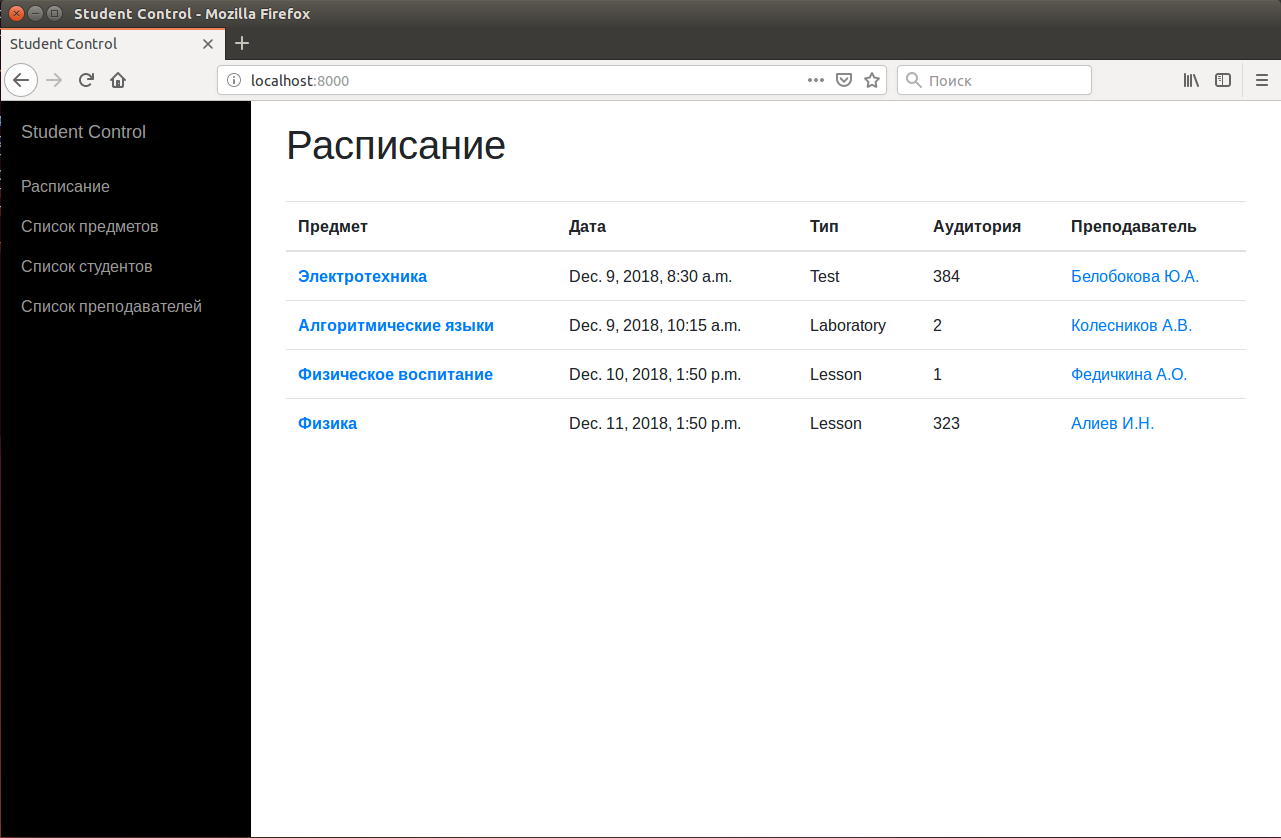
**

*Рис. 2. Запуск сервера*

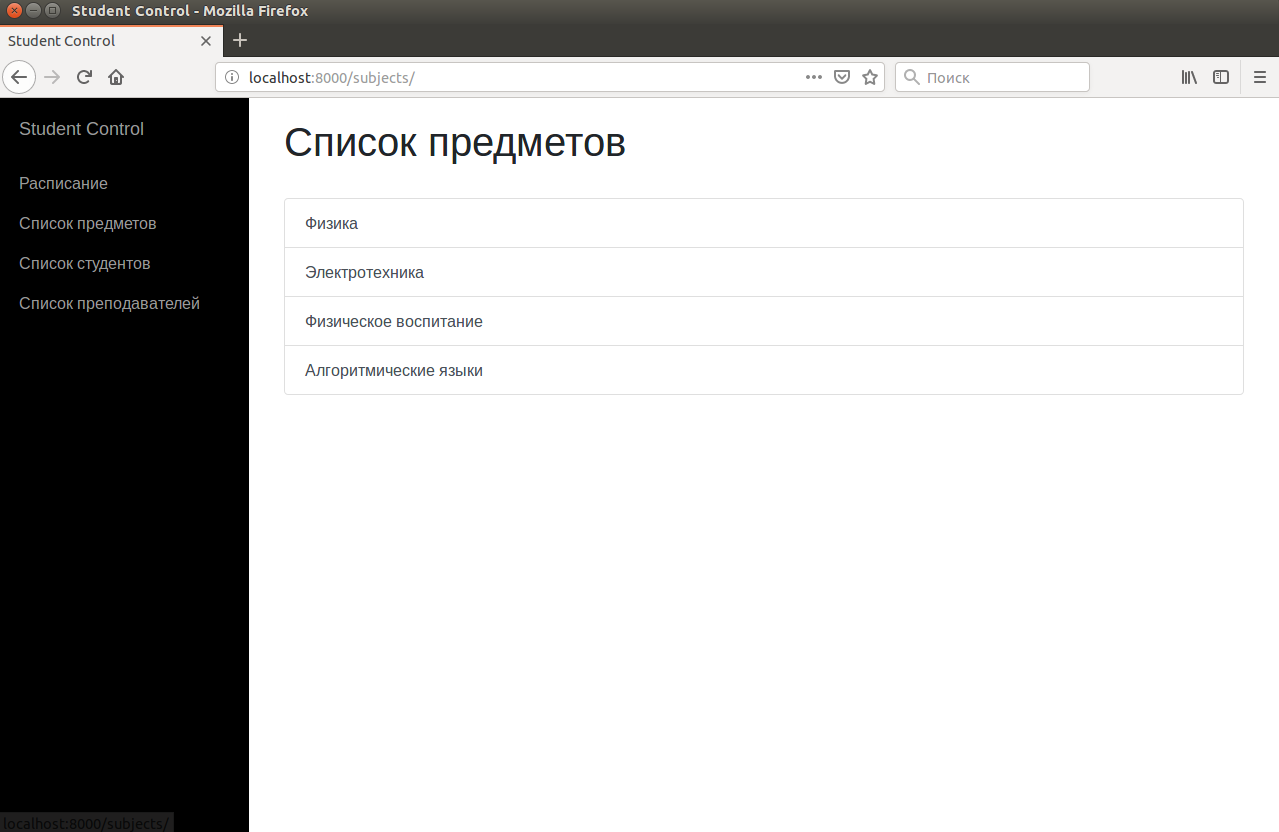
В браузере перейти по адресу *localhost:8000* для просмотра информации, или *localhost:8000/admin/* для ввода или изменения информации.

# *7.2 Экранные формы*

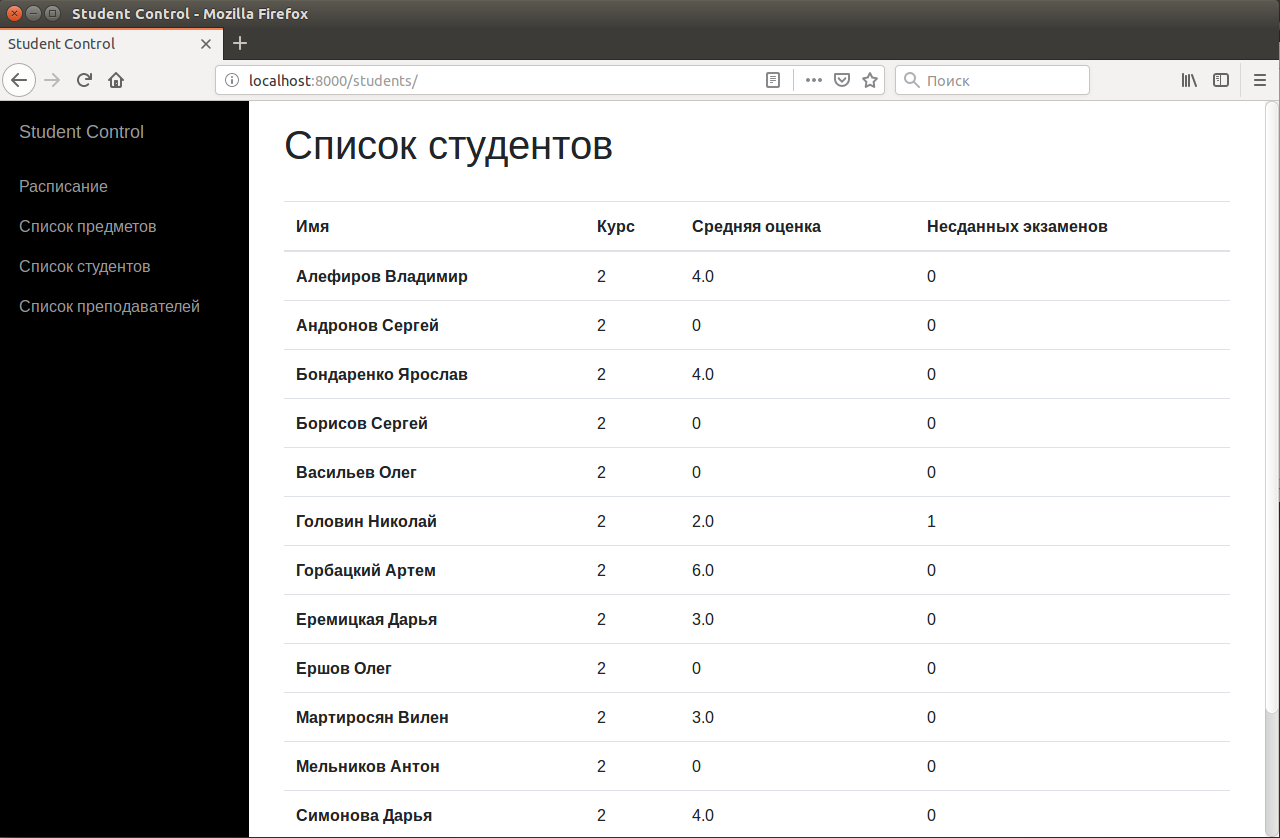
### *7.2.1 Экранные формы пользователя студент*



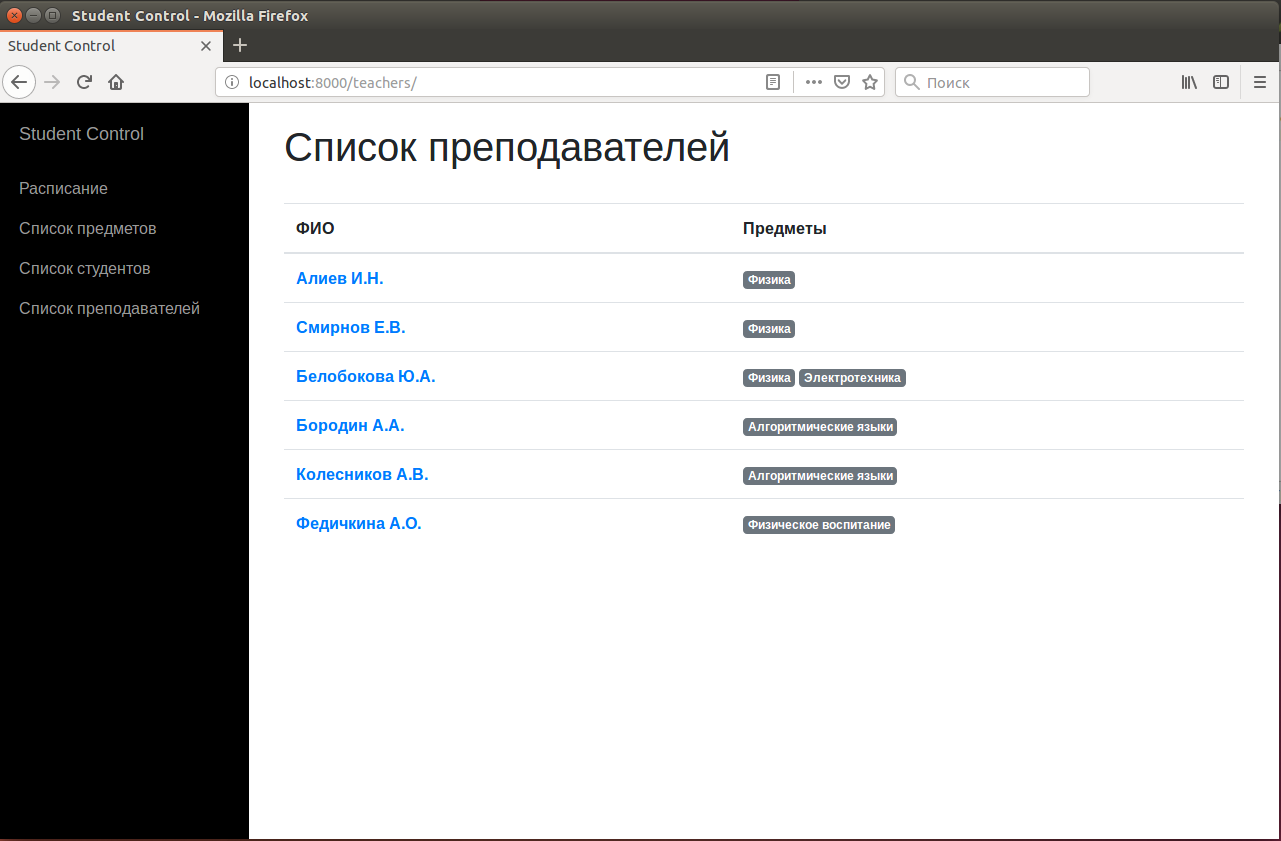
*Рис. 3. Экранная форма – “Основная страница”*



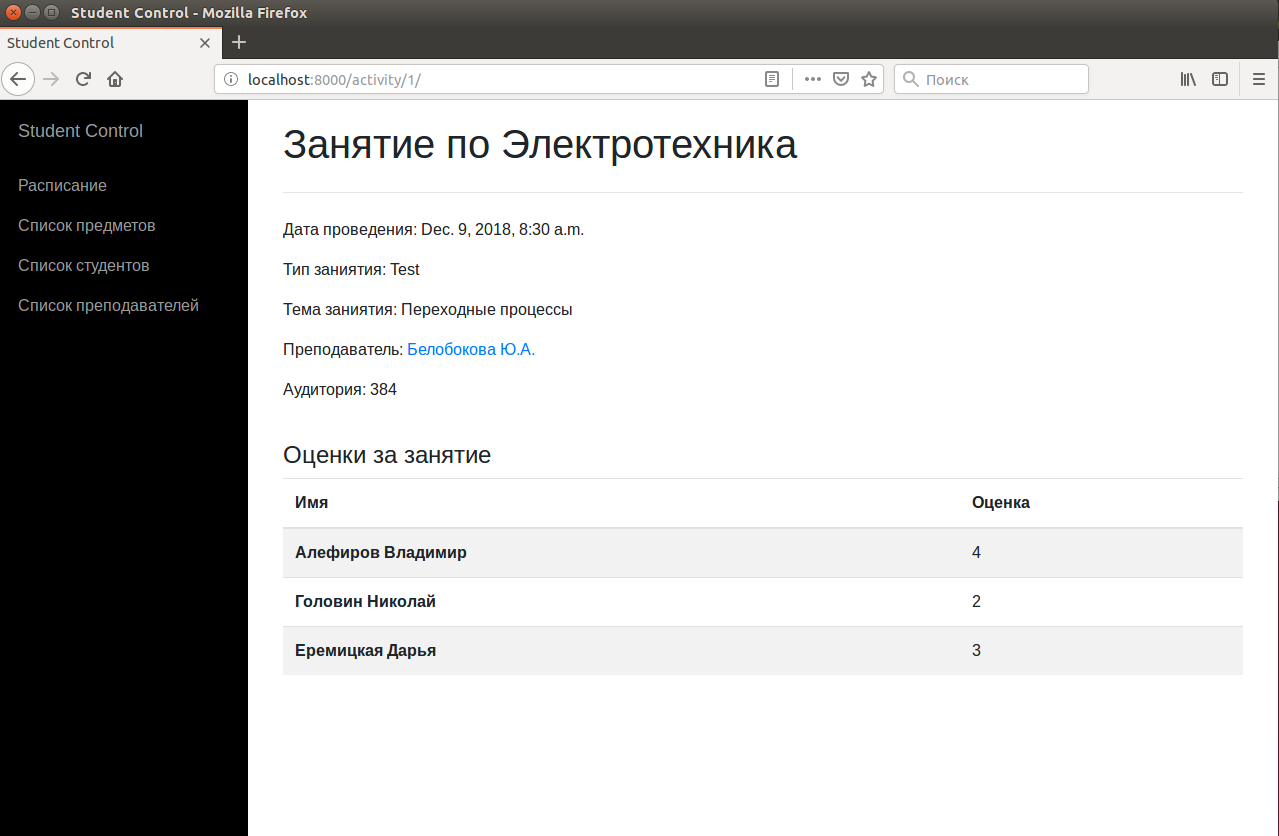
*Рис. 4. Экранная форма – “Список предметов”*



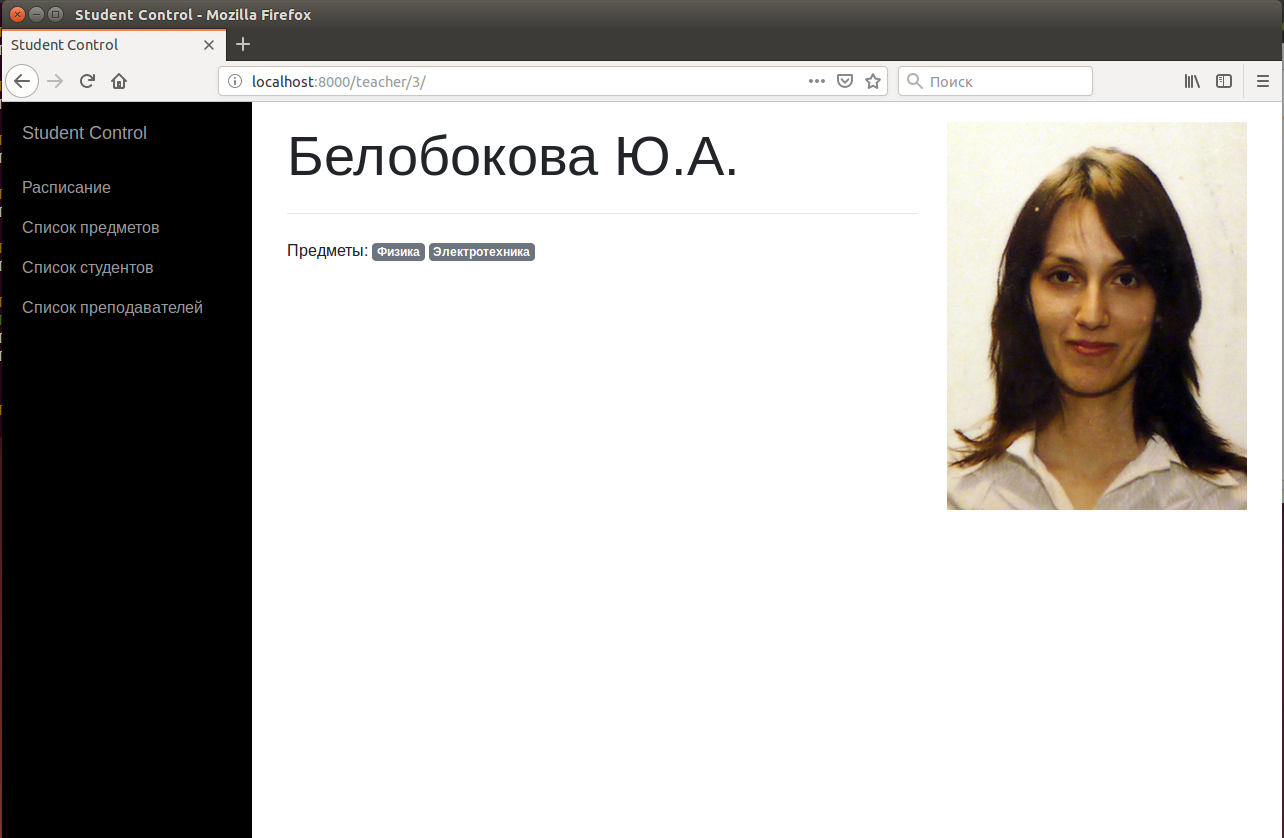
*Рис. 5. Экранная форма – “*Список студентов*”*



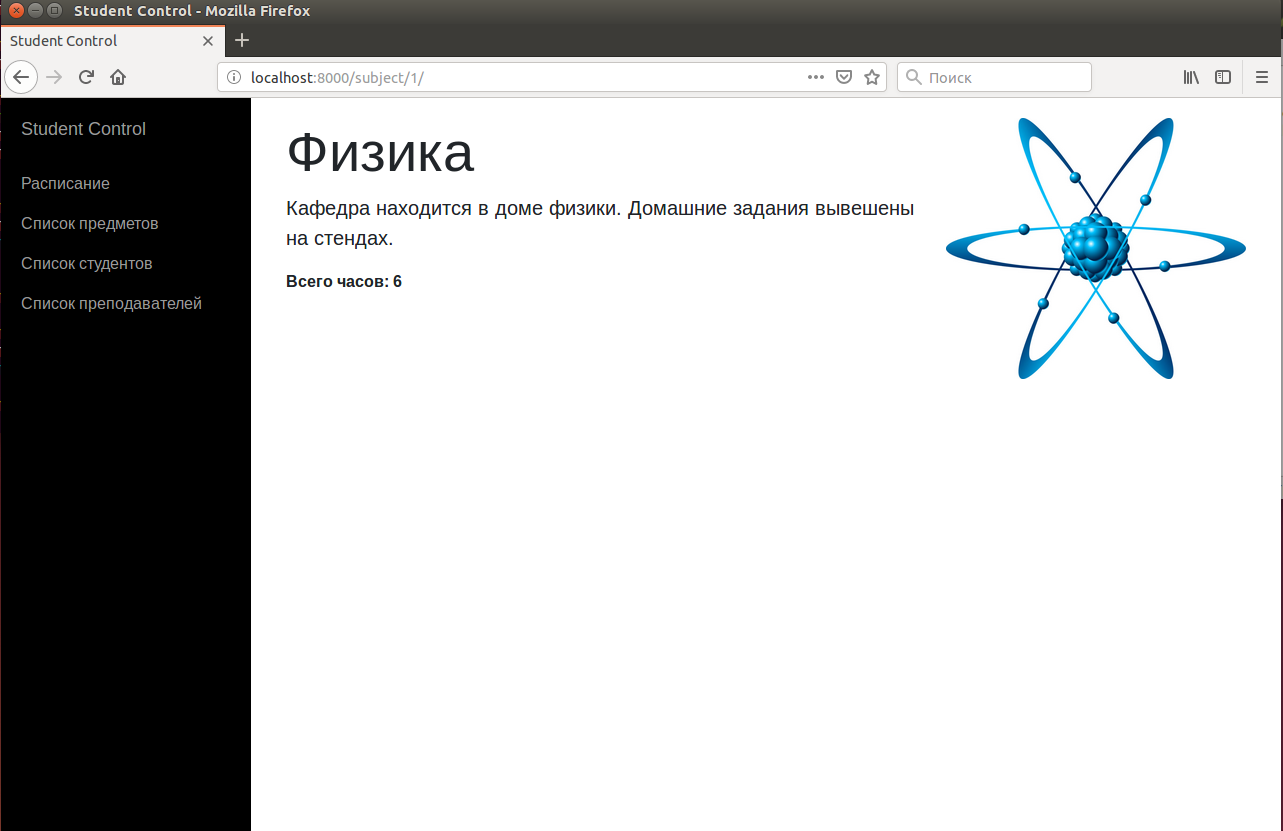
*Рис. 6. Экранная форма – “*Список преподавателей*”*



*Рис. 7. Экранная форма – “Информация о занятии”*



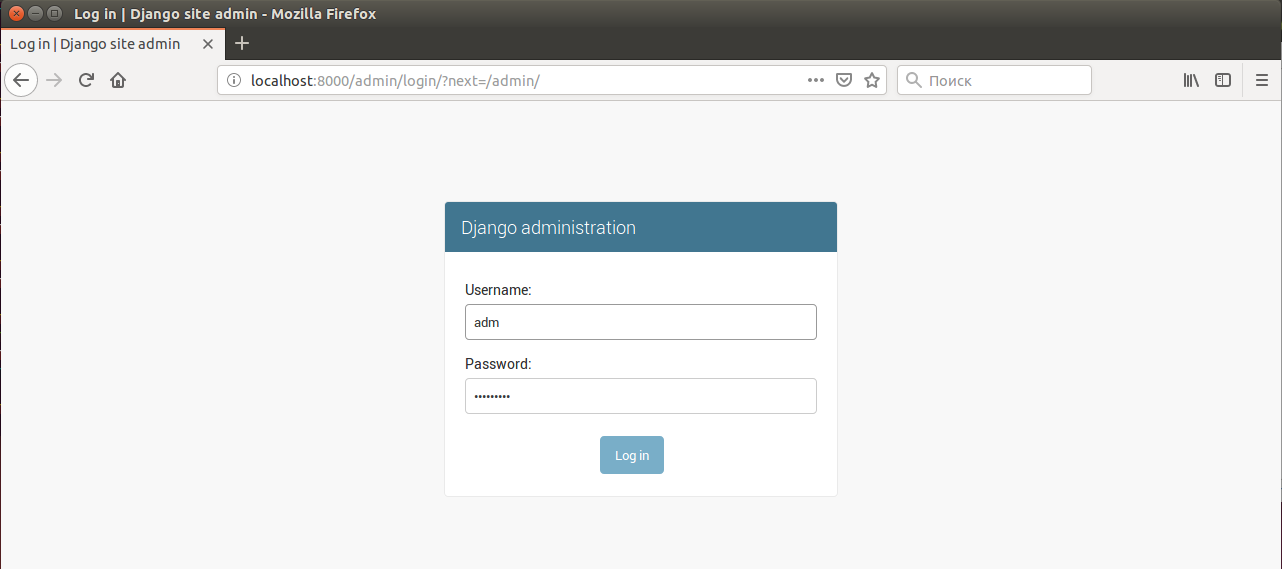
*Рис. 8. Экранная форма – “Информация о преподавателе”*



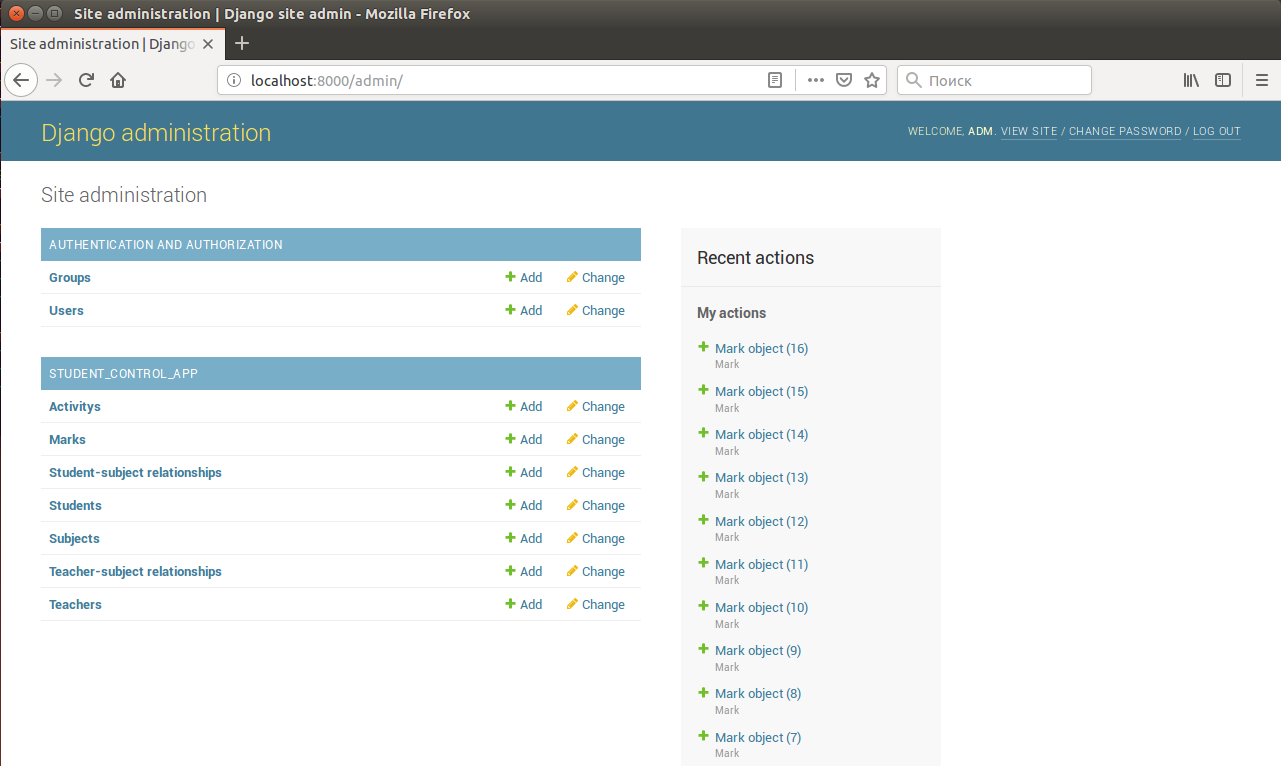
*Рис. 9. Экранная форма – “Информация о предмете”*

# 

### *7.2.2 Экранные формы пользователя преподаватель*

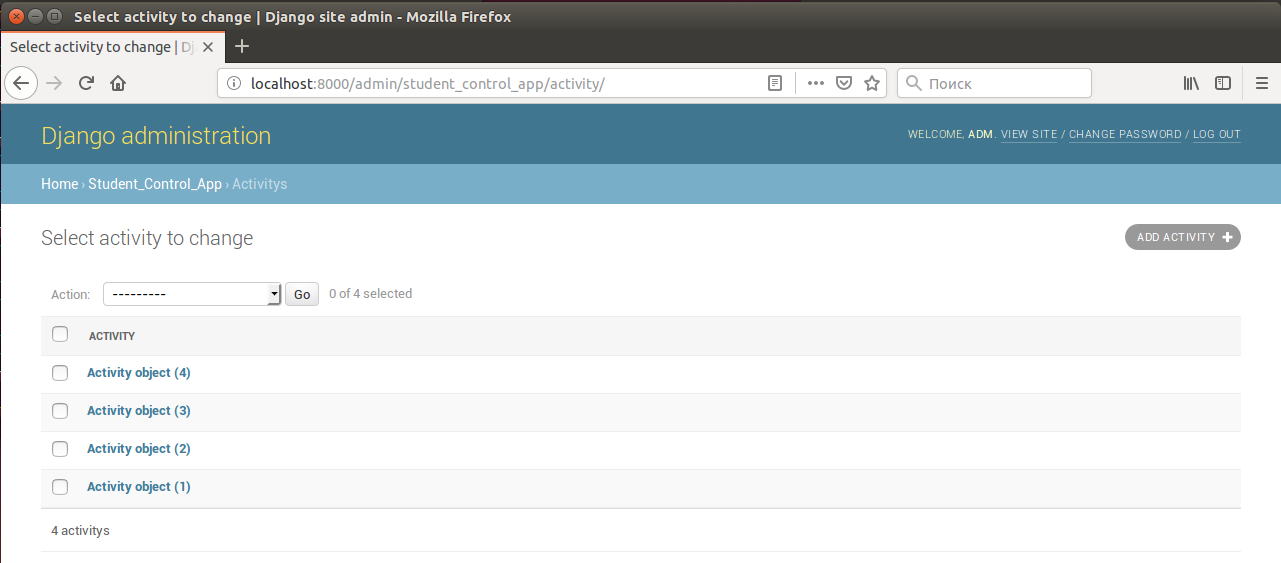


*Рис.10. Экранная форма – “Вход в панель администратора”*

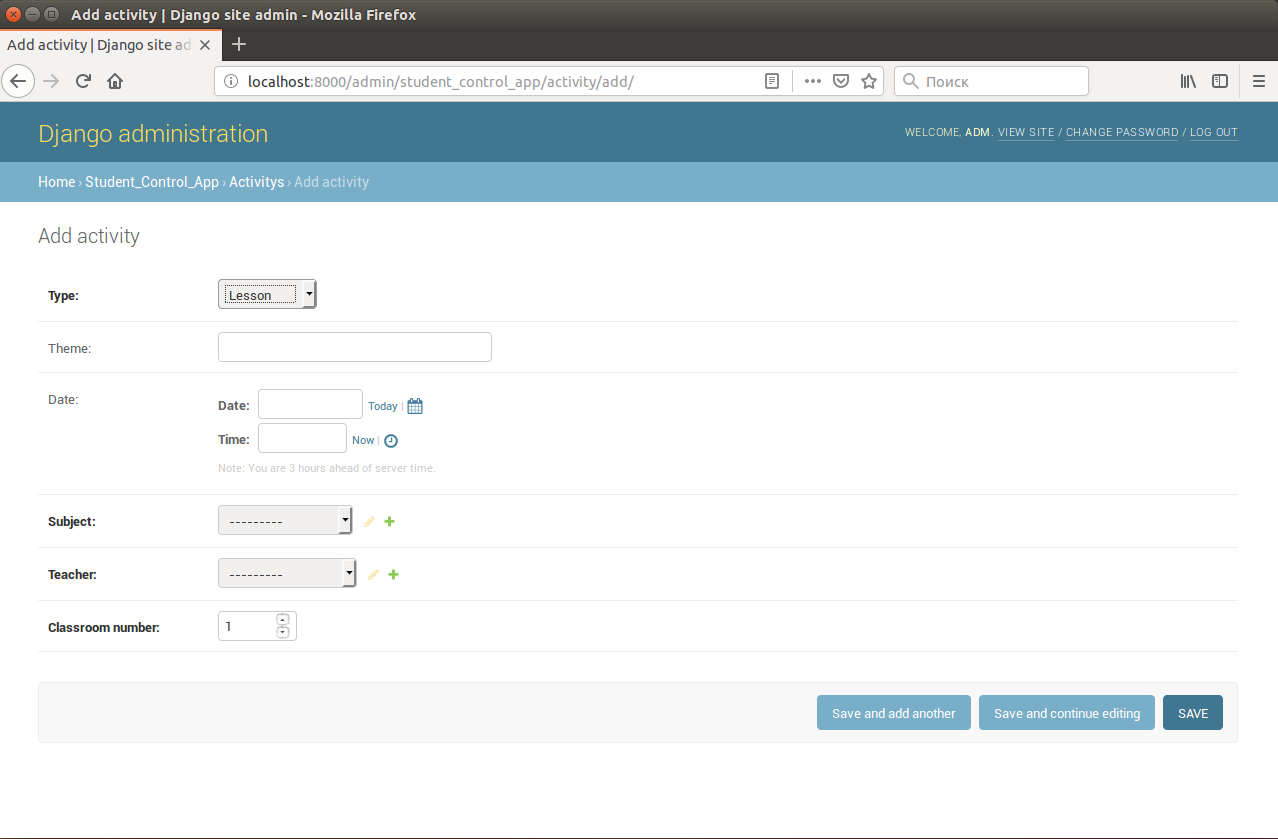


*Рис. 11. Экранная форма – “Вход в панель администратора”*

Для изменения или добавления информации нажать на соответствующее значение в списке

**

*Рис. 12. Экранная форма – “Список всех записей занятий”*



*Рис. 13. Экранная форма – “Добавление нового занятия”*

## 7.3 Добавление новой информации

Для добавления новой записи необходимо:

* Нажать кнопку 
* Заполнить все необходимые поля
* Нажать кнопку SAVE

# 8. Руководство пользователя

|  |  |  |  |
| --- | --- | --- | --- |
| N п. | Исходное состояние | Действие | Ожидаемый результат |
| 1. | Окно браузера | Переход по адресу *localhost:8000* | Переход на главную страницу – «Расписание» |
| 2. | Страница расписание | Нажать на занятие | Вывод страницы информации о занятии |
| 3. | Страница занятия | Нажать на преподавателя | Страница преподавателя |
| 4. | Страница преподавателя | Нажать на предмет | Страница предмета |
| 5. | Любая страница сайта | Нажать на «Список предметов» | Вывод списка предметов |
| 6. | Страница списка предметов | Нажать на предмет | Страница предмета |
| 8. | Любая страница сайта | Нажать на «Список преподавателей» | Вывод списка преподавателей |
| 9. | Страница списка преподавателей | Нажать преподавателя | Страница преподавателя |
| 10. | Любая страница сайта | Нажать на кнопку «Список студентов» | Вывод списка студентов и информации о них |
| 11. | Окно браузера | Переход по адресу *localhost:8000/admin/* | Страница ввода логина и пароля |
| 12. | Страница «﻿*Site administration»* | Нажать на кнопку «+ Add» | Добавление нового объекта |
| 13. | Страница «﻿*Site administration»* | Нажать на кнопку «Change» | Изменение объекта |

*Таб. 1. Руководство пользователя.*

# 9. Заключение

В процессе выполнения курсовой работы были достигнуты поставленные цели:

**- были получены навыки инфологического проектирования баз данных**

**- была освоена СУБД SQLite**

**- были получены навыки создания веб-приложений к базам данных на Python**

**- был получен навык верстки веб-страниц**

**- были получены навыки грамотного оформления документации: описана предметная область; составлена инфологическая модель; разработана DFD диаграмма**

Разработанная система является удобной для учета успеваемости студентов. Студентам она позволяет просматривать расписание, информацию о занятиях, информацию о преподавателях и предметах. Так же студенты могут видеть рейтинг по среднему баллу и количество своих несданных экзаменах. Преподавателям система позволяет легко изменять расписание, добавлять и удалять студентов, ставить оценки за занятия. Ведение учета в электронном виде позволит избежать кропотливой возни с разрозненными документами.

Систему можно изменять и дорабатывать в процессе использования, что сделает её более профильной и «заточенной» под решение определенных задач.