Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

**Отчет по лабораторной работе №1**

Дисциплина: «Разработка профессиональных приложений»

**Обработка списков**

Вариант №4

Выполнил:

студент группы ИВТАСбд-22

Данилин Н.Г.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

[Ульяновск, 2023](https://github.com/ghub-ayrtom/DPA-lab5)

[**Постановка задачи**](https://github.com/ghub-ayrtom/DPA-lab5)

**Вариант задания:**

4. Финансовые переводы: №, дата и время, сумма, описание транзакции

Для данного варианта задания необходимо реализовать веб-сайт в среде Django, удовлетворяющий следующим требованиям:

1. Количество таблиц в БД - не менее 5;

2. Должны быть реализованы 3 типа связей: один-к-одному, один-ко-многим, многие-ко-многим;

3. Таблицы создаются с использованием моделей в Django и миграций;

4. Как минимум, для 1 сущности должны быть реализованы CRUD операции (Create, read, update, delete) + представление списка сущностей;

5. Для остальных сущностей добавить только средства редактирования;

6. Необходимые пять таблиц должны быть разработаны для моделирования предметной области, заданной вариантом. Необходимо придумать эти таблицы, не забывая про здравый смысл.

7. Добавить минимум 2 роли, соответствующих реальным ролям в выбранной предметной области. Добавить пользователей, обладающих этими ролями.

8. Реализовать средства регистрации пользователей, аутенфикации.

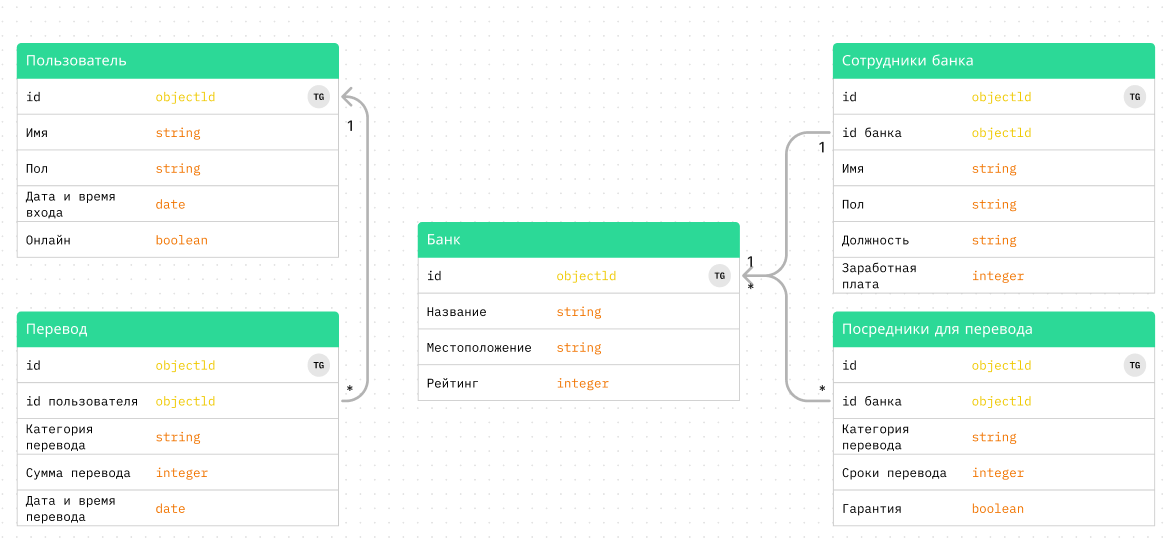
9. Добавить средства авторизации и разграничения прав доступа разным ролям к разным сущностям.

**Описание реализации**

Для начала, в файле models.py, в котором хранятся все сущности базы данных, были придуманы и добавлены ещё 4 таблицы, которые моделируют предметную область, заданную моим вариантом:

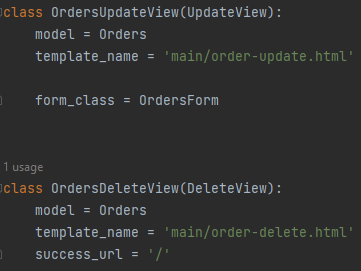


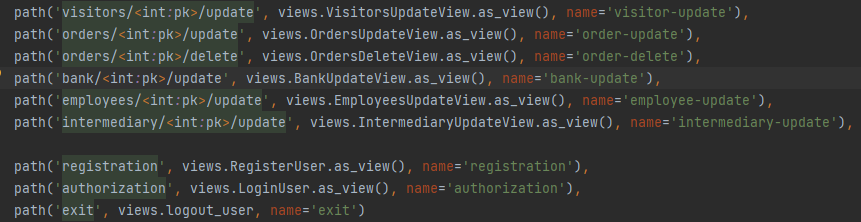
Для реализации типа связи один-к-одному в таблице Сотрудники в качестве первичного ключа (ID) было использовано поле OneToOneField, для реализации же типа связи многие-ко-многим в таблице Посредники в качестве ID было использовано поле ManyToManyField с указанием таблицы для связи.

Для добавление CRUD операций в таблицу Переводов пользователей банка, а также средств для редактирования всех сущностей, были созданы специальные «умные» формы в файле forms.py, которые могут считывать и записывать информацию в базу данных Django:

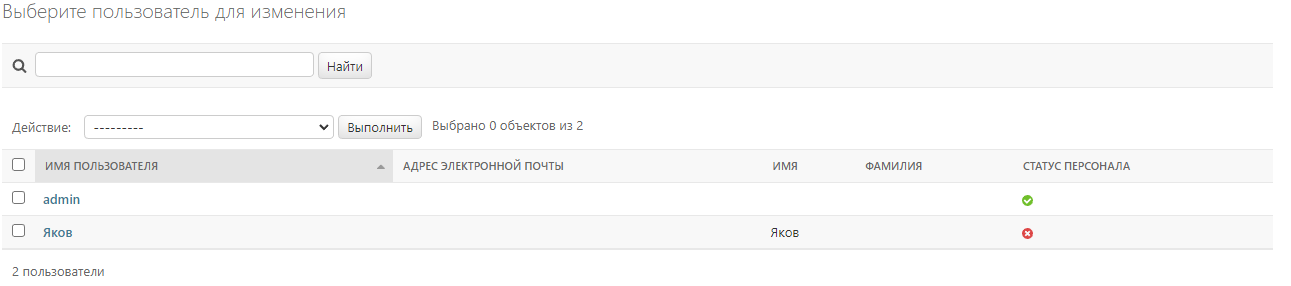


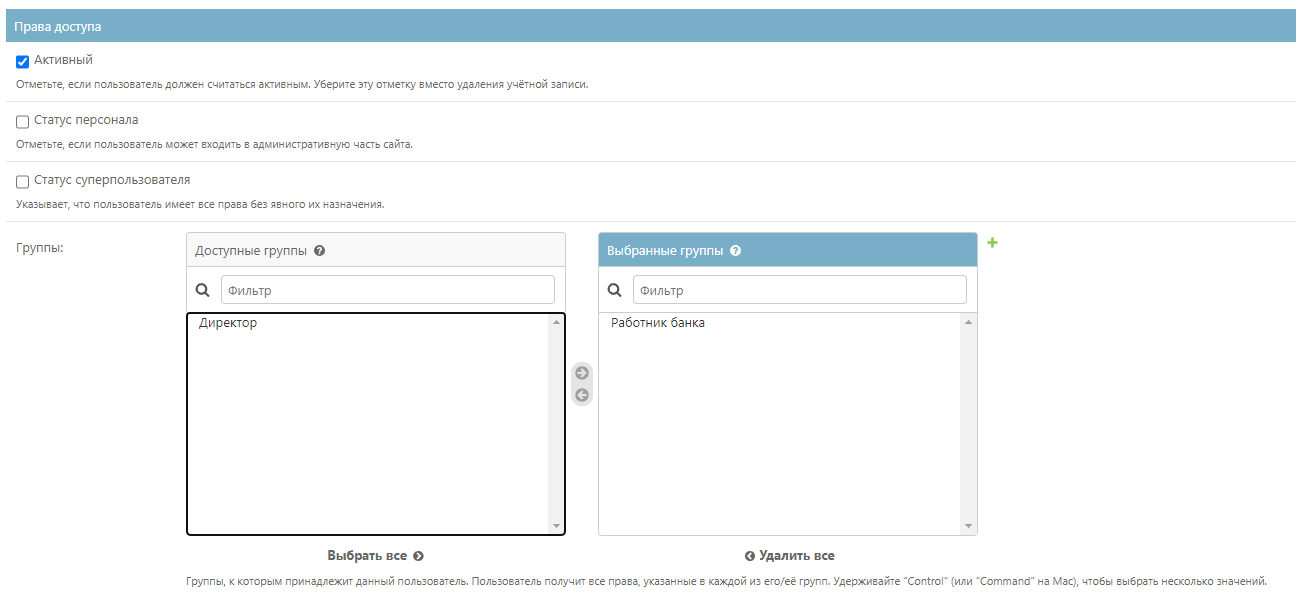
Для их связи с базой данных, а также вёрсткой, в файле views.py были созданы специальные View-классы, а переходы (нажатия на кнопки Добавления и Удаления) отслеживались в файле urls.py:

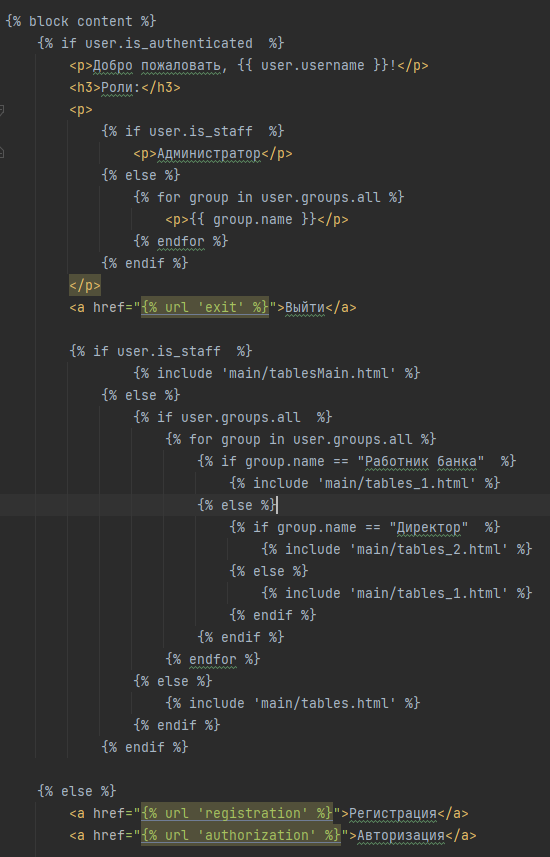


 Также, в панели администратора Django, были добавлены 2 роли: Работник магазина и Рекрутёр. Все они соответсвуют реальным ролям в выбранной предметной области:

После этого, с использованием всё тех же «умных» форм, View-классов, а также URL-переходов были реализованы средства для регистрации и авторизации пользователей, так как, по-сути, таблица пользователей в базе данных Django ничем не отличается от добавленных ранее сущностей, и работа с ней осуществляется по тем же правилам:

 После того, как пользователь был зарегистрирован, ему через панель администратора можно присвоить одну из добавленные ранее ролей и при его следующем входе в свой аккаунт, его роль будет считана и отображён соответсвующий шаблон вёрстки только с теми сущностями, к которым данная роль имеет доступ. Таким образом, были реализованы средства разграничения прав доступа разным ролям к разным сущностям:





**Описание возникших затруднений**

Основные затруднения при выполнении данной лабораторной работой были связаны с добавлением функционала разграничения прав доступа разным ролям к разным сущностям, так как при этом, необходимо считывать роль из базы данных Django и отображать соответствующий шаблон вёрстки. Также, было непросто реализовывать такие типы связей как один-к-одному и многие-ко-многим, так как до этого мне не приходилось с ними сталкиваться.

**Вывод**

По итогу выполнения данной лабораторной работы я разработал веб-сайт в среде Django, который удовлетворяет всем требованиям, описанным в постановке задачи, а его предметная область соответствует моему варианту задания