Разработать приложение для управления потоками людей в медицинском учреждении. В системе есть три группы пользователей: пациенты, врачи и администраторы. За основу взять дефолтную таблицу User и расширить необходимыми полями.

У любого пользователя есть:

* ФИО
* Пол
* Дата рождения
* Группы, в которых он состоит (пациент, врач, администратор)

У врача дополнительно задаётся:

* Время работы на каждый день недели (пн, вт, ср, чт, пт, сб, вс) Какие-то дни могут быть пустыми (суббота и воскресение, например).
* Количество дней, на которое генерируются слоты для записи на приём (по умолчанию, на 7 дней вперёд).
* Также для врачей должна быть возможность задать отпуск или перерыв в работе на обед. Формально, это диапазон datetime. Для этого промежутка времени генерировать слоты не нужно!
* Время обслуживания у конкретного врача = длины слота. Например, 20 минут.

К примеру, у врача рабочий день начинается в 11:00 и заканчивается в 13:00. Время обслуживания 20 минут. Тогда необходимо сгенерировать 6 слотов: 11:00-11:20, 11:20-11:40, 11:40-12:00, 12:00-12:20, 12:20-12:40, 12:40-13:00. Генерация должна быть автоматической через celery.

Для ускорения разработки уточню поля Timetable:

* id
* doctor – врач, внешний ключ на расширенную модель пользователя
* client – пациент, внешний ключ на расширенную модель пользователя
* start\_time – дата и время начала слота
* stop\_time – дата и время окончания слота

Необходимые методы API:

1. Создать пациента или врача с общими данными (ФИО, пол, дата рождения), установить группы пользователя. Входные данные: id пользователя. Необходимые права: администратор.
2. Изменить данные пациента или врача. Входные данные: id пользователя. Необходимые права: администратор.
3. Заблокировать пациента или врача. Входные данные: id пользователя. Необходимые права: администратор.
4. Получение свободных слотов у врача в будущем. Входные данные: id врача. Выходные данные: id слота, время начали и время окончания слота. Необходимые права: любой авторизованный пользователь.
5. Возможность записи на приём к врачу (заполнение слота). Входные данные: id слота. ID клиента узнать по токену. Необходимые права: пациент. Обратите внимание, что надо проверять, свободен ли слот. Если да, то выдавать ошибку.
6. Возможность отмены записи на приём. Необходимые права: пациент.
7. Получение статистических данных: сколько записей на какой день. Пример: {“day”: “2021-07-19”, “count”: 10, “day”: “2021-07-20”: “count”: 5. Крайне желательно использовать группировать данные при запросе к БД. Перебор всех объектов из БД в цикле for – не лучшее решение.
8. Метод авторизации. Входные данные: логин и пароль. Выходные данные: токен (необязательно JWT).

Требования к стеку технологий:

* Python 3.6 и выше
* Django 2.1 и выше & DjangoRestFramework
* PostgreSQL
* Celery
* Pytest
* Docker
* Docker-compose

Итого:

1. Расширить таблицу User
2. Разработать модель Timetable для слотов
3. Разработать автогенерацию слотов
4. Разработать 8 методов API. Реализовать можно в несколько строк кода: задать queryset, serializers. проверку прав.
5. Написать несколько юнит-тестов к разработанным функциям
6. Обернуть решение в docker-compose

На решение даётся 2-3 дня. Код можно выложить на github или gitlab. Если что-то не успели, то ничего страшного =). Успехов в решении. Do your best!