

Задача №1. Архитектура БД

Легенда: Туристическая компания “Star Travel” занимается организацией отдыха клиентов в пределах одной страны. В услуги компании входит бронирование перелета в место назначения, отеля для проживания и транспорта для перевозки клиентов от аэропорта до отеля. Компании потребовалась CRM система, которая будет отвечать следующим бизнес запросам:

1. Возможность у менеджера создавать, изменять, удалять клиентов в системе. Хранить информацию о ФИО клиента, дате рождения, паспортных данных (серия, номер, дата выдачи, кем выдан), способы связи с клиентом (номер телефона, email, телеграм)
2. Для клиента должен быть обязательно определен хотя бы один способ связи, допускается указание 2-х и более способов (телефон email или email телеграм).
3. Менеджер должен иметь возможность создавать и изменять программы туров, несущие в себе следующую информацию: место проведения (указывается город), краткое название, полное описание программы.
4. Возможность менеджера создавать, изменять, удалять бронирования туров для клиентов. Бронирование состоит из выбора программы тура и заказанных услуг. В рамках работы с бронированиями клиентов менеджер должен иметь возможность заполнить данные о заказанных услугах: бронирование отеля, бронирование перелета, бронирование транспорта.
5. Для каждого заказа клиента обязательно бронируется транспорт, а бронирование отеля и перелета - по желанию клиента. Менеджер должен иметь возможность отметить в системе, что клиент отказался от бронирования отеля или перелета. В этом случае данные об услуге заполнять не надо.
6. Бронирование транспорта включает в себя как минимум 2 события: трансфер от аэропорта до отеля по прилете и от отеля до аэропорта в конце тура. Менеджер должен иметь возможность организовать дополнительные бронирования для клиента по его желанию. Бронирования транспорта должны включать в себя следующие данные: автомобиль(модель, марка, гос номер, фио водителя), место прибытия, время прибытия.
7. Бронирование отеля может отсутствовать или быть только одно в рамках купленного клиентом тура. При заказе этой услуги, менеджер должен заполнить следующую информацию: указать отель (название, адрес, email), указать номер (комната в отеле), дату приезда, дату отъезда.
8. Бронирование авиа перелета может отсутствовать или быть заполнено. Авиаперелет подразумевает под собой бронирование двух билетов - целевой и обратный билет. Клиент может заказать: целевой и обратный, только целевой, только обратный или отказаться вообще от услуги. В рамках каждой брони менеджер должен иметь возможность заполнить следующие данные: указать авиаперевозчика, указать номер билета, зарегистрированного на клиента, указать дату отправления рейса.

Уведомления

1. После создания бронирования менеджером, клиенту должно прийти уведомление в соответствии с выбранным способом связи (если указано несколько способов связи, то отправить на каждый из способов соответствующее уведомление). На каждую услугу - отдельное уведомление.
2. Для каждого способа связи (email, телефон, telegram) должны существовать шаблоны, которые менеджер может изменять. В шаблоне должна содержаться информация о названии шаблона, к какому способу связи относится шаблон, текста шаблона (в тексте шаблона содержатся переменные, необходимые для подстановки значений).
3. Все отправленные уведомления должны быть залогированы. В логах должна быть следующая информация: клиент, способ связи, выбранный шаблон, финальный текст после подстановки переменных в шаблон, дата отправки.
4. При изменении каких либо данных в бронировании должна осуществляться отправка уведомления по вышеописанному алгоритму.
5. Менеджер должен иметь возможность просматривать список всех отправленных уведомлений по каждой из услуг выбранных клиентом в рамках его бронирования тура.

Аналитика

В системе должна быть представлена следующая аналитика:

1. Популярность и сезонность программ туров - количественный (общий график и график по каждой программе тура) на временной шкале, отражающий, сколько раз программа тура была заказана клиентами
2. График отправленных уведомлений с временной шкалой, где каждая линия - это способ связи. График должен отражать количество отправленных уведомлений, агрегированных по дням/месяцам/годам.

Задача:

1. Разработать концептуальный уровень представления структуры БД (сущности, связи, кратности) на основе ТЗ.
2. Представить результат в виде ER-диаграммы

Задание №2. DRF

Легенда: У вас на руках есть проект “Публицист”, в котором авторы могут публиковать свои статьи. Анонимные участники могут комментировать и ставить оценки статьям, при этом система определяет, с какого типа устройства производилась активность: с мобильной или веб-версии. Также стоит помнить, что у каждой статьи может быть один или более тегов, а может и не быть вовсе. По желанию автор может заполнить о себе дополнительную информацию: телефон или email.

Вам предоставляется исходный код проекта, ER-диаграмма архитектуры БД и описание заданий.

Предварительная подготовка

1. Скачайте архив **test_task_middle.zip** и разархивируйте

2. По желанию можете развернуть проект на своей рабочей машине, для этого следуйте инструкции из **README**
3. После выполнения тестового задания пришлите вновь заархивированный проект или загрузите на git и пришлите ссылку на репозиторий

Задание: Фильтры

эндпоинт `/api/articles/` (Список статей) маршрутизируется на `ArticleListView`. В параметрах GET запроса можно указать ключ `category`. За логику фильтрации отвечает класс фильтра `ArticleFilter`

Задача: изменить методы в `ArticleFilter` таким образом, чтобы в зависимости от значения `category` в параметрах запроса результат соответствовал ожидаемым требованиям. Требования установлены в описании к соответствующему методу

Задание: Аннотации

эндпоинт `/api/articles/stats` (Список статей со статистикой) маршрутизируется на `ArticleListStatsView`.

Задача: изменить метод `get_queryset` в `ArticleListStatsView` таким образом, чтобы кроме всех полей модели `Article` были ещё аннотированы следующие поля:

`count_comments` = Количество комментариев у статьи

`count_ratings` = Количество оценок у статьи

`average_rating` = Средний рейтинг у статьи

`author_age` = Возраст автора на текущую дату

`author_age_in_publish` = Возраст автора на момент публикации статьи

`count_activities` = Общее количество активностей у статьи: комментарии + оценки

Не забудьте раскомментировать соответствующие поля в `ArticleListStatsSerializer`