

Задание №1: Онлайн-аукцион

Aryslan Yakshibaev

12 сентября 2021 г.

Содержание

1	Задание	1
2	Функциональные требования	1
3	Роли пользователей и действия для каждой роли	2
3.1	Роли пользователей	2
4	Объекты, о которых будут храниться данные	2
5	Связи между объектами для хранения данных	2
6	Схема объектной модели	3

1 Задание

Вам поручено разработать онлайн-аукцион. Он позволяет продавцам продавать свои товары с помощью аукциона. Покупатели делают ставки. Выигрывает последняя самая высокая ставка. После закрытия аукциона победитель оплачивает товар с помощью кредитной карты. Продавец отвечает за доставку товара покупателю.

- ☒ Предложите список функциональных требований для проекта.
- ☒ Определите роли пользователей и действия для каждой роли.
- ☒ Определите объекты, о которых будут храниться данные.
- ☒ Определите связи между объектами для хранения данных.
- ☒ Нарисуйте схему объектной модели (используя любые обозначения, которые вам удобны).

2 Функциональные требования

Так как сервис будет использоваться как для покупки, так и для продажи, потребуется пользовательская сущность, **Пользователь**. Также, учитывая что это предполагается публичный сервис, потребуются дополнительно роли **администратора** и **модератора**, у модераторов должно быть право удалять любой неприемлемый контент и блокировать пользователей-нарушителей, а администратор должен иметь те же права, что и модераторы, плюс управление ролью модератора, т.е. давать и забирать эту роль у пользователей.

Регистрация и вход в систему должны быть по электронной почте и паролю, у каждого пользователя также должно быть имя и, опционально, аватарка.

Также необходимы сущности **Аукцион** и **Ставка**. Допустим, что для продажи одного товара нужно создать один аукцион, т.е. "один аукцион-один товар". Пусть у каждого аукциона будет опциональное дата-время окончания (timestamp). Конечно же, должна быть возможность устанавливать заголовок, фото, описание, стартовую цену и дедлайн для аукциона. Каждый пользователь кроме владельца аукциона может ставить, но каждая новая ставка должна быть больше

текущей максимальной, либо \geq стартовой цене, установленной продавцом. Также продавец должен иметь возможность назначить минимальную дельту для следующей цены (т.е. минимальную разницу между текущей максимальной и новой ставками).

Считаю, что также требуется хотя бы минимальный функционал комментирования, чтобы пользователи могли оставлять комментарии к аукционам.

3 Роли пользователей и действия для каждой роли

3.1 Роли пользователей

- **Пользователь** может:
 - Просматривать все ставки
 - Просматривать все аукционы
 - Редактировать свои ставки
 - Редактировать свои аукционы
 - Просматривать данные всех пользователей
 - Редактировать свои данные (профиль)
 - Просматривать все комментарии
 - Оставлять комментарии к аукционам, редактировать или удалять свои комментарии
- **Модератор** может:
 - Всё, что может пользователь
 - Удалять любые аукционы
 - Банить любых пользователей, которые ниже его по роли
- **Администратор** может:
 - Всё, что может модератор
 - Выдавать и забирать роль модератора у пользователей

4 Объекты, о которых будут храниться данные

- Аукцион
- Ставка
- Asset (в данном случае фото/видео, приложенные к аукциону). Для хранения предлагается использовать SQL blob.
- Комментарий (пользователя к аукциону)
- Пользователь

5 Связи между объектами для хранения данных

См. схему

- Пользователь – Аукцион: one-to-many
- Ставка – Аукцион: many-to-one
- Ставка – Пользователь: many-to-one
- Аукцион – Asset: one-to-many
- Аукцион – Комментарий: one-to-many
- Asset – Пользователь: many-to-one
- Пользователь – Комментарий: one-to-many

6 Схема объектной модели

Ссылка на схему в PonyORM

(Так как PonyORM редактор не поддерживает некоторые типы данных, такие как *blob* и *enum*, я обозначил эти типы в названии полей)