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.

1. **Функциональные требования:**

***Общие требования:***

* Должен существовать личный кабинет продавца. Он должен зарегистрироваться и запомнить свой пароль. Должна быть функция восстановления пароля.
* Должен существовать личный кабинет покупателя. Он должен зарегистрироваться и запомнить свой пароль. Должна быть функция восстановления пароля.
* Каждый покупатель должен видеть список доступных сейчас аукционов (описание товара и текущая ставка) .

***Процесс аукциона:***

* Должна быть возможность выставлять любой товар на аукцион (для продавца, в его личном кабинете)
* Наличие товара у продавца должно быть подтверждено (с помощью фото или видео тоже в личном кабинете продавца при попытке выставить товар на аукцион).
* Также, продавец (при попытке выставить товар на аукцион) подтверждает свои намерения с помощью заполнения договора, обязывающего в случае подтверждения оплаты от покупателя выслать товар по нужному адресу.
* Должна быть возможность поставить начальную ставку (для продавца, в его личном кабинете, при подтверждении его намерения продать товар)
* Покупатель должен суметь войти на страницу аукциона и принять в нем участие при подтверждении своих намерений с помощью заполнения договора. Договор обязывает выплатить нужную сумму продавцу в случае победы или отказаться от оплаты и приобретения товара.
* У покупателя должна быть возможность сделать любую ставку, которая **выше текущей.**
* Программа не должна принимать ставки одновременно (то есть из тех, кто выставил 2 одинаковые ставки *почти* одновременно, выбирается тот покупатель, который поставил раньше). При этом покупателю, которому в ставке отказано, должно прийти какое-нибудь оповещение об отказе.
* Программа должна отсчитывать время, за которое можно сделать следующую ставку. Если это время вышло, выявляется победитель – последний поставивший. Аукцион закрывается и всем приходит оповещение о закрытии аукциона. Победителю приходит оповещение о победе и инструкция по оплате.

***Процесс оплаты и доставки:***

* Победивший покупатель должен оплатить товар с помощью своей кредитной карты по указанным реквизитам, которые приходят к нему в личный кабинет от продавца.
* Никакая личная информация победившего покупателя не утекать (не должна быть доступной ни для кого, кроме продавца)
* Как только продавцу в личный кабинет приходит подтверждение оплаты, он должен узнать от покупателя адрес для доставки.
* Как только продавец получает адрес и отправляет товар, покупателю в личный кабинет приходит оповещение о том, что товар отправлен.
* Как только покупатель получает товар, продавцу в личный кабинет приходит оповещение о том, что товар доставлен.

***Критические ситуации:***

* Если фото или видео товара не загружено продавцом, или он отказывает подписать договор, ему не должна быть дана возможность выставить товар на аукцион
* Если покупатель отказывается подписывать договор, ему не должна быть дана возможность участия в аукционе
* Если покупатель в случае победы отказывается от оплаты, продавцу должно прийти соответствующее оповещение. Тогда, продавец может выставить на аукцион товар еще раз или не выставлять.
* Если оплата от покупателя не получена, продавцу должна предоставляться возможность связаться с покупателем напрямую для выяснения всех обстоятельств.
* Если товар покупателем не получен, покупателю должна предоставляться возможность связаться с продавцом напрямую для выяснения всех обстоятельств.

1. **Роли пользователей и действия для каждой роли:**

Как было описано в функциональных требованиях, роли и действия следующие:

* Продавец

Действия:

- выставить товар на аукцион (+сделать необходимые подтверждения такие как, снять фото или видео, подписать обязательства),

- поставить начальную ставку,

- отправить победившему покупателю реквизиты счета,

- подтвердить оплату от покупателя,

- получить адрес для доставки от покупателя,

- отправить оповещение о том, что товар отправлен,

- получить подтверждение о том, что его товар получен

- связаться с покупателем в случае критической ситуации (см функциональные требования)

* Покупатель

Действия:

- выбрать доступный аукцион из списка

- повысить ставку

- получить отказ в ставке, если кто-то его опередил

- видеть время, оставшееся для повышения ставки

- получить оповещение о закрытии аукциона

- получить оповещение о победе

В случае победы:

- получить реквизиты продавца

- отправить адрес доставки

- получить оповещение об отправлении

- отправить оповещение о получении товара

- связаться с продавцом в случае критической ситуации (см функциональные требования)

1. **Объекты, о которых будут храниться данные:**

* Продавец (Seller)
* Покупатель (Customer)
* Аукцион (Auction)
* Объект, в котором хранится информация о взаимодействии покупателя с аукционом
* Объект, в котором хранится информация о взаимодействии покупателя с продавцом

1. **Связи между объектами:**

* Seller – Customer (many to many)

То есть, как продавец может взаимодействовать со многими покупателями, так и наоборот

Или можно сказать, что ***{Seller} – {Customer – Seller Interaction) is one-to-many***

И ***{Customer} – {Customer – Seller Interaction) is one-to-many***

* Seller – auction (one to many)

То есть продавец может делать несколько аукционов, но у каждого аукциона фиксированный один продавец

Получаем*:* ***{Seller} – {Auction} is one-to-many***

* Customer – auction (many to many)

То есть, покупатель может участвовать во многих аукционах, и в одном аукционе может участвовать много покупателей

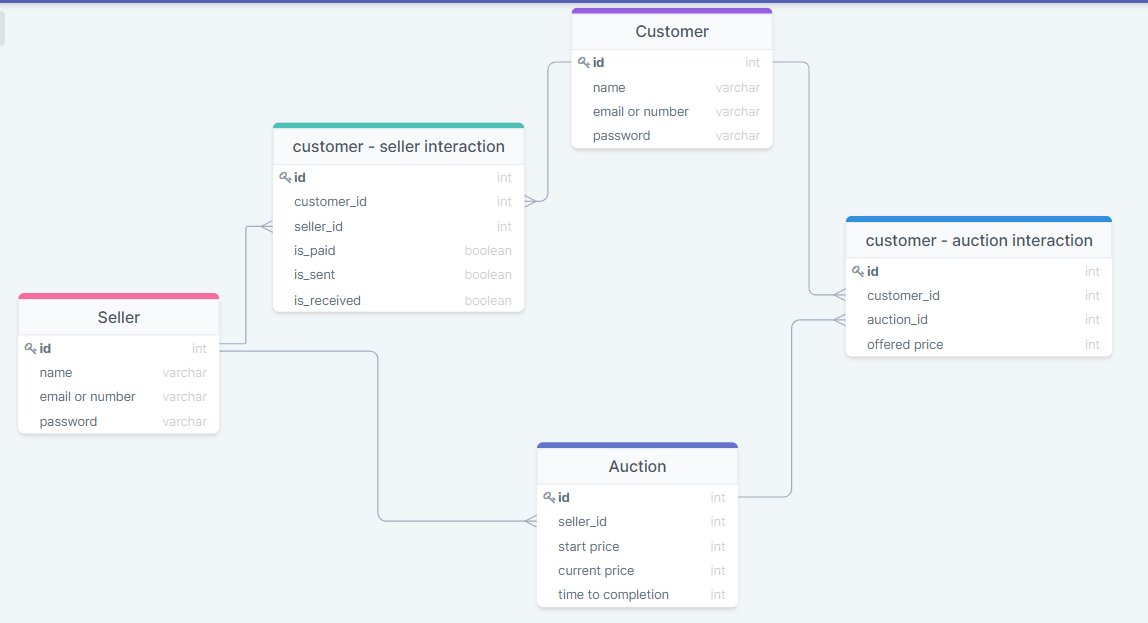
Или же, можно сделать через взаимодействие покупателя и аукциона:

***{Customer} – {Customer – Auction Interaction} is one-to-many***

И ***{Auction} – {Customer – Auction Interaction} is one-to-many***

1. **Схема объектной модели**

Схема, которую я нарисовала, находится ниже в виде картинки или по ссылке <https://drawsql.app/higher-school-of-economics/diagrams/task1#>



*Схема объектной модели*