ТЗ CRM

СRM система нужна для приема заказов с разных источников. Для этого нужно сделать апи чтобы можно было удобно добавлять заказы из разных площадок (магазинов).

Вкратце объясню архитектуру и сценарий нашей работы для понимания что мы хотим от црмки:

У нас есть онлайн магазины по продаже БАДов (и наши и аффилиатов)

Покупатель попадает на магазин, кладет товар в корзину и нажимает кнопку Checkout. Его редиректит на страницу биллинга (она одна для всех магазинов) с инфой откуда он пришел и что положил в корзину. На биллинге он заполняет биллинг инфо (оплата у нас картой, пайпал, криптой, чеком) и шиппинг инфо и нажимает оплатить.

Платежные данные попадают на процессинг оплаты, и вся инфа о заказе к нам в црм.

Далее идет обработка заказа. От мерчанта приходит коллбек к нам в црм и статус ордера меняется. Уходят письма покупателю в зависимости от статуса.

Шиппинг также через нашу црмку (отправка заказов поставщику, отправка трек номеров).

Также должен быть аккаунт оператора колл центра для обзвона заказов с неудачной оплатой и решений вопросов с клиентом.

—--------------------------------

**Какие данные мы получаем когда клиент делает заказ?**

Клиент, после выбора товаров, попадает к нам на биллинг для заполнения паймент,шиппинг информации с такими JSON данными:

{

"data": {

"product: ["

{

"id": 64764, //Product id из магазина

"sku": "ОЕК345", //Product SKU из магазина. Это код товара, по которому мы идентифицируем какой продукт из базы поставщика отправлять клиенту.

"name": "VITAMIN B", //Название продукта

"option": "20 mg",

"option\_value": "30 pills",

"bonus": "4 pills",

"quantity": 1,

"price": 20, //цена в той валюте которая была на чекауте магазина

"url": "https://site.com/product",

"thumbnail": "https://site.com/img/product.png"

},

{...}, // Больше продуктов

],

"coupon": {

"used": 1, // 0 - не применен, 1 - применен

"type": 1, //в магазине два типа: percentage (значение 1) и fixed amount (значение 2)

"code": "XMAS24", // код купона

},

"shipping": {

"id": 1, // 1 - AirMail, 2 - EMS

"price": 14.95, //цена доставки

"insurance": 0, //insurance это опция гарантированной повторной доставки в случае потере посылки, 0 - не применен, 1 - применен

"insurance\_price": 4.95

},

"shop": {

"id": 1, //статичное значение условного id магазина

"hostname": "site.com", //адрес сайта магазина

"order\_id": "HFT677875"

},

"total": 34.95, // тотал всей корзины

"currency\_coef": 0.95, //коэффициент пересчета от основной валюты, по которому магазин пересчитывал и получил Total цену

"discountedTotal": 15, //тотал с учетом скидки примененного купона

"totalProducts": 1, // кол-во продуктов в корзине

"currency": "EUR", //ISO код валюты

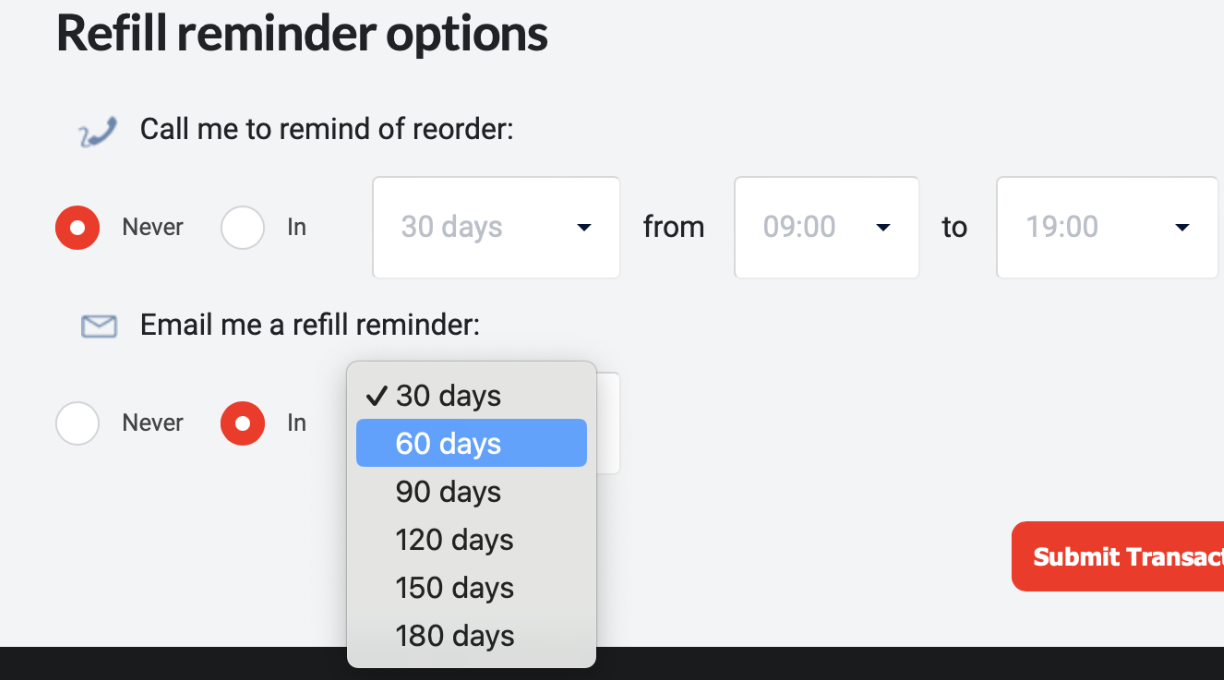
"language": "ENG" //язык который был выбран на магазине

}

}

**Далее клиент на биллинге заполняет такие данные:**

* First name\*
* Last name\*
* Email\* (email является идентификатором клиента, по нему мы сможем понять это новый покупатель, или он уже покупал у нас)
* Mobile phone\*
* Address\*
* Address 2
* Country\*
* State\* (только для стран у которых есть штаты, иначе отправляется XX)
* Zip\*
* Payment method\* (название либо айди любого метода оплаты который мы добавим на биллинг).
* Credit card number (если выбрана оплата картой)
* Credit card expiry date (если выбрана оплата картой)
* Credit card CVV (если выбрана оплата картой)
* Credit card holder name (если выбрана оплата картой)
* Birth date (если выбрана оплата картой)
* Refill reminder options\* (выбор напоминания о повторном заказе, через какой период времени напоминать и каким методом связи, нужно для обзвона клиентов). Включает в себя 2 варианта выбора, на скрине наглядно:



—-----------------------------------------

**Взаимодействие с API**

1. Обмен данными должен осуществляется по протоколу SSL методом POST, данные отправляются в формате JSON, закодированных в BASE64. Данные приходят в значении параметра **data** в теле POST запроса.

Пример запроса

$requestData = [

'auth' => [

'api\_id' => '105',

'api\_key' => '5e31872b64a5820bc2c32914adcf232b',

],

];

В теле POST запроса будет data=JHJlcXVlc3REYXRhID0gWwogICAgICAgICAgICAgICAgJ2F1dGgnID0+IFsKICAgICAgICAgICAgICAgICAgICAnYXBpX2lkJyA9PiAnMTA1JywKICAgICAgICAgICAgICAgICAgICAnYXBpX2tleScgPT4gJzVlMzE4NzJiNjRhNTgyMGJjMmMzMjkxNGFkY2YyMzJiJywKICAgICAgICAgICAgICAgIF0sCiAgICAgICAgICAgIF07Cg==

1. Авторизация

Для авторизации должны использоваться auth.api\_id и auth.api\_key.

auth.api\_id - ID магазина в системе

auth.api\_key - API ключ от магазина

Авторизационные данные должны присутствовать в любом запросе к API.

Это всего лишь пример от нас, делайте так как вы считаете будет лучше!

**Оформление заказа предлагаю сделаю так:**

URL: https://нашдомен/create\_order.php

Запрос:

auth - объект авторизации

auth.api\_id - ID магазина

auth\_api\_key - ключ магазина

customer - объект покупателя (Все поля обязательные)

customer.first\_name - имя покупателя

customer.last\_name - фамилия покупателя

customer.email - email покупателя **(ЭТО БУДЕТ ИДЕНТИФИКАТОР ПОВТОРНО КУПИВШЕГО КЛИЕНТА)**

customer.phone - телефон клиента

customer.ip - IP клиента

billing - объект биллинга (Все поля кроме штата обязательны)

billing.country - ISO-2 код страны

billing.state - ISO-2 код штата или XX если не US

billing.city - город покупателя

billing.zip - индекс покупателя

billing.address - адрес покупателя

shipping - объект шиппинга (Все поля кроме штата обязательны)

shipping.country - ISO-2 код страны

shipping.state - ISO-2 код штата или XX если нету штата

shipping.city - город покупателя

shipping.zip - индекс покупателя

shipping.address - адрес покупателя

payment - объект оплаты

payment.method - id платежного метода (Обязательно) - У нас есть такие методы оплаты - Credit Card, PayPal, Cryptocurrency, Echeck

payment.cardholder - владелец карты

payment.card\_number - номер карты

payment.card\_expire\_month - месяц истечения карты (2 знака)

payment.card\_expire\_year - год истечения карты (4 знака)

payment.card\_cvv - CVV карты

payment.cardholder\_birthday - дата рождения в формате ММ-ДД-ГГГГ

options - объект опций заказа

options.delivery\_id - ID способа доставки (Обязательно)

options.insurance - Гарантия доставки (Необязательно)

options.currency - ISO-3 код валюты (Обязательно)

options.locale - Используемая локаль (Необязательно)

options.campaigns - UTM-метка. (Необязательно)

options.shop\_domain - Домен который будет указан в письме (Необязательно)

options.shop\_order\_id - ORDER ID магазина который будет указан в письме (Обязательно. При наличии shop\_order\_id параметр shop\_domain обязателен)

options.bonus - бонус товара (как выше "bonus": "4 pills") выбранного в качестве бонуса. (Необязательно)

metrics - объект мета-информации о заказе (Необязательно)

metrics.device - объект информации об устройстве (Необязательно)

metrics.device.user\_agent - User Agent (Необязательно)

metrics.device.screen\_width - ширина экрана (Необязательно)

metrics.device.screen\_height - высота экрана (Необязательно)

metrics.device.screen\_avail\_width - доступная ширина (Необязательно)

metrics.device.screen\_avail\_height - доступная высота (Необязательно)

metrics.device.screen\_color\_depth - глубина цвета экрана (Необязательно)

metrics.device.language - язык системы (Необязательно)

metrics.device.timezone - часовой пояс (Необязательно)

products - массив состава заказа (Обязательно)

В начале в json я писал как передается инфа о продуктах.

**Ответ:**

result - результат операции (true / false)

error - текст ошибки или пустота

id - ID созданного заказа у НАС в системе. Нужно создавать какой-то захешированный уникальный айдишник, к примеру в sha256 options.shop\_order\_id + customer.email+unix timestamp. Подумайте как лучше..)

Заказы будут передаваться в разных валютах, но **вся статистика в црмке должна рассчитываться в долларах.** Но сами заказы и действия с ними конечно должны храниться и производиться в той валюте с которой были созданы.

—-----------------------------------------

**Статусы заказов**

Всем покупателям выставляются статусы при определенных логических условиях (например когда мы отправляем трекинг номер то заказ получает статус SHIPPED).

Также после выставления статуса должно идти письмо покупателю, с персонализированными данными. Тексты писем мы поправим потом сами, вы сделайте пока сам функционал.

Для отправки писем используем внешний смтп.

**Могут быть такие статусы заказов после поступления в црм с последующими действиями:**

1. Pending

Как только заказ попал в систему, он под этим статусом (или если с оплатой заказа пока не понятно) -> Письмо что вы сделали ордер.

1. Approved

Оплаченый заказ. -> Письмо что ваш ордер успешно оплачен и готовится на отправку. -> Добавляется сумма на баланс с вычетом процента в зависимости какой процессинг оплаты был использован для заказа.

1. Declined

Оплата не прошла. -> Письмо

1. Cancelled

Клиент отменил заказ -> Письмо

1. Call Center

Заказ на обзвоне в колл центре.

1. Packed

Заказ в службе доставки, но еще не отправлен. -> Письмо

1. Shipped

Заказ отправлен, трекинг номер отправлен. -> Письмо с треком

1. Reshipped.

Посылка отправлена повторно, так как не дошла первая. -> Письмо с треком

1. Delivered.

Это если прикрутим авто отслеживание доставки по трек номеру. -> Письмо

1. Refunded.

Деньги возвращены. -> Письмо -> Отнимаются деньги с баланса.

1. Chargeback.

Опасные кастомеры, которые звонят в банк и говорят что они не оплачивали товар, банк делает чарджбек и возвращает им деньги принудительно со счета продавца. -> Письмо -> Отнимаются деньги с баланса.

1. Closed

Закрытый заказ, оплата не прошла и связаться не удалось. -> Письмо

—--------------------------------------------

По структуре БД не буду расписывать, проектируйте на свое усмотрение, с учетом всего написанного в этом ТЗ. Добавлю только:

1. Данные кредитной карты как-то под хешем хранить что ли, но мы можем повторные транзы по запросу клиента делать, поэтому их нужно во время отправки на платежный процессор расшифровывать, в общем подумайте как лучше это сделать.
2. В БД должен храниться id платежа. На нашей стороне есть id заказа в нашей системе, а когда мы отправляем заказ на процессинг оплат, то он отдает id платежа/id инвойса в своей системе, по которому платеж идентифицируется на его стороне.
3. В БД должно храниться название процессинга через который прошла оплата заказа. У нас “каскад” методов оплаты (процессингов) и у каждого свое апи для взаимодействия.
4. В БД должно храниться название/айдишник поставщика, через которого будет отправлена посылка, так как их тоже несколько и у каждого разный метод приема информации о заказах.

—-------------------------------------------

**Фронтенд**

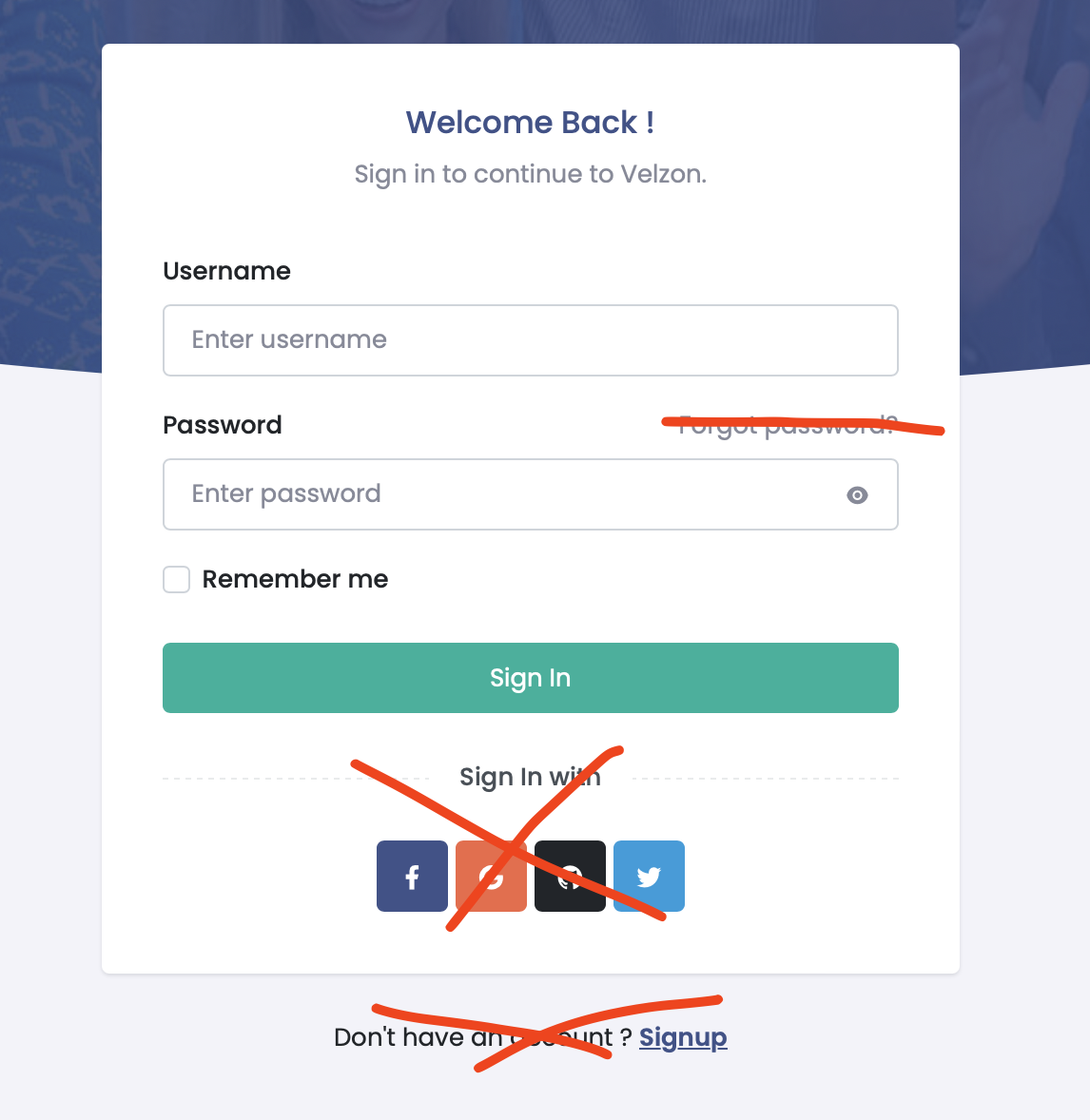
Делаем на шабе что я вам скинул. Все скрины буду брать из того что есть в шаблоне.

**Login page**

Рега нам не нужна.

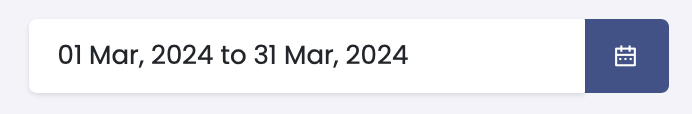
Remember me сделать куку подольше, на месяц если можно.

Всего будет 2 аккаунта. Менеджер заказов и Оператор колл центра (далее объясню).



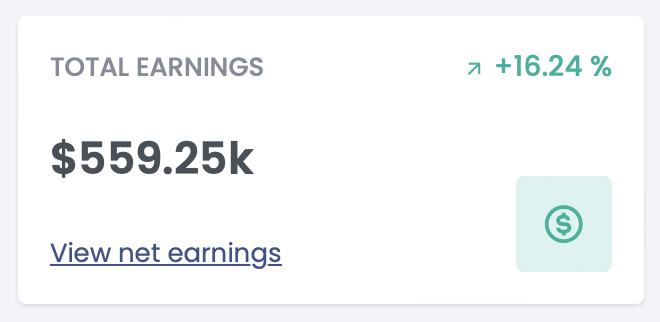
—-----------------------------------------

**Dashboard**

Отображать инфу по умолчанию за текущий месяц. Но оставить выбор диапазона дат 

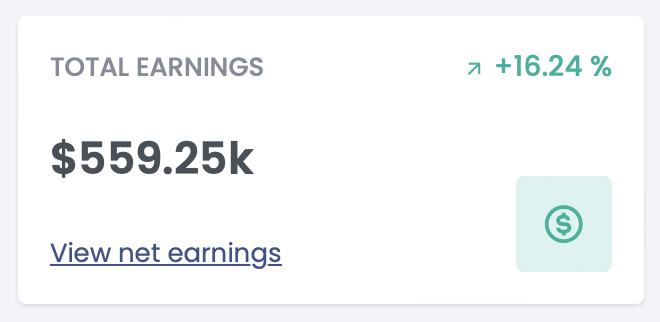
Виджеты которые нужны на дашборде:

1. Всего заработано грязными (TOTAL SALES). Это сумма всех ордеров которые зашли, вне зависимости от статуса.

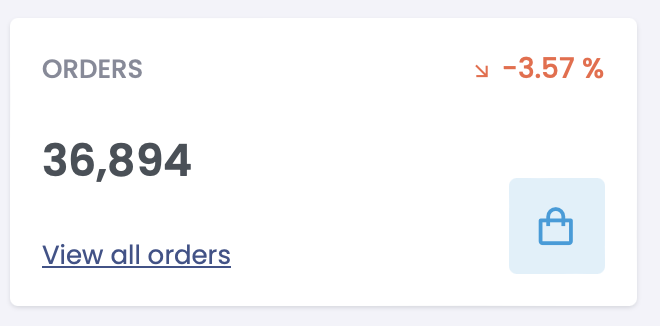


Там где  можно не отображать.

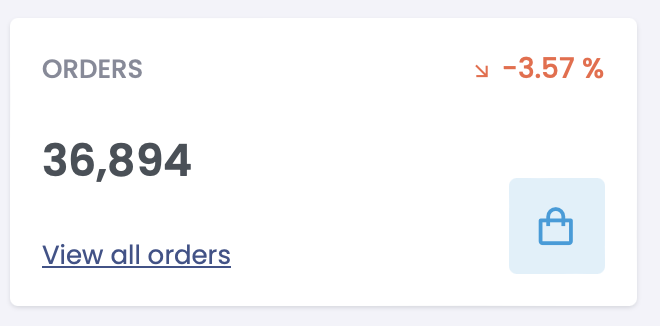
1. Всего заработано чистыми (NET EARNINGS). Это ордера только со статусами успешной оплаты и последующими SHIPPED и тп.



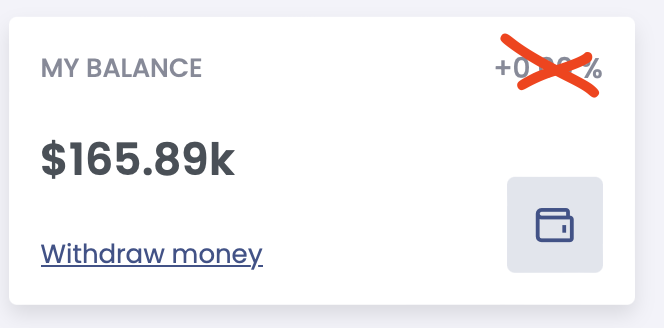
1. Количество заказов



1. Количество оплаченых заказов (APPROVED ORDERS)



1. Баланс. Сюда будет добавляться наш чистый профит - это не вся сумма с заказа. Допустим заказ был на 100$, но процессинг оплаты берет комиссию 10%, наш профит составит 100$-10%=90$.

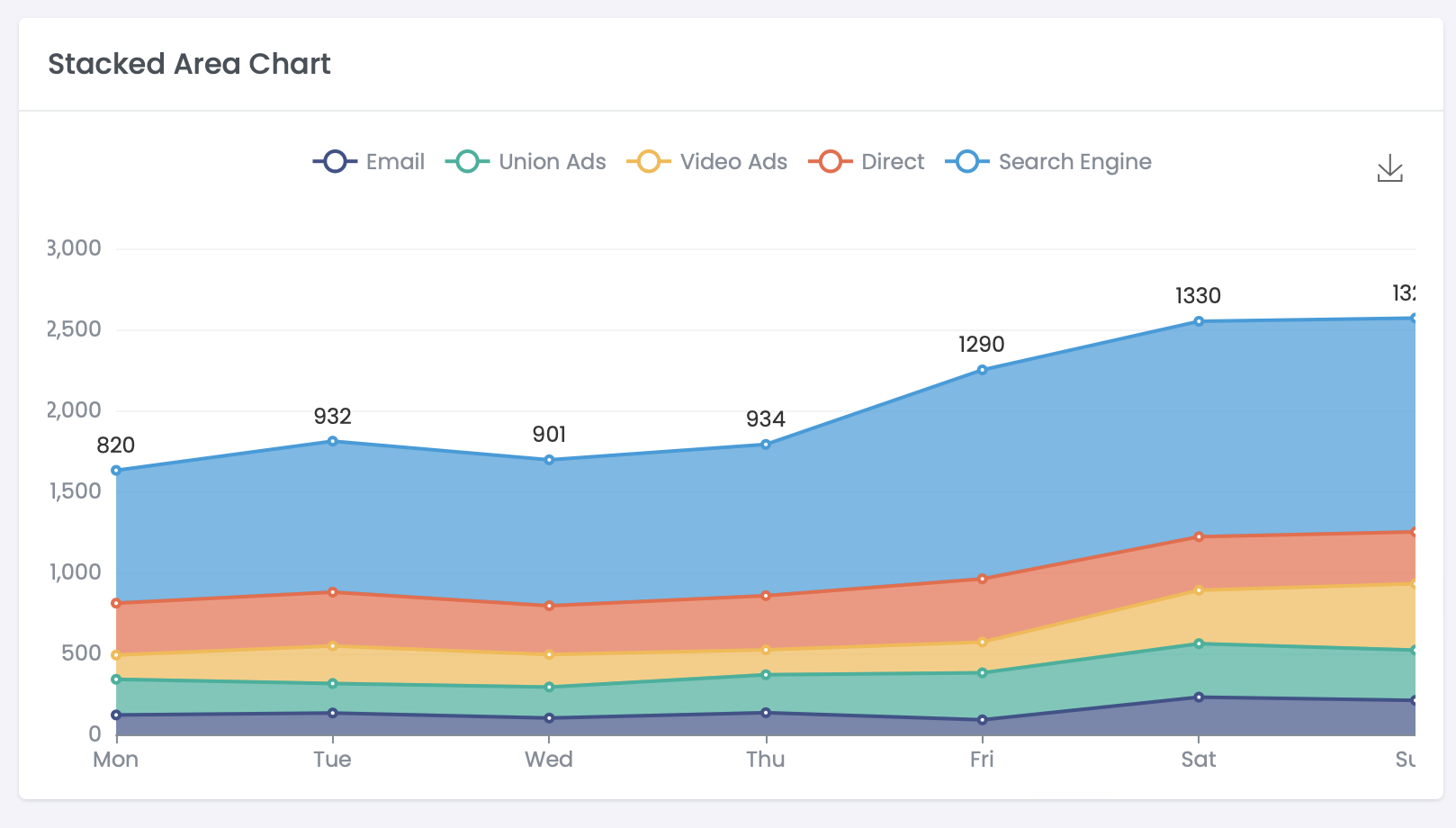


1. Turnover by days.Оборот по трем статусам заказов за каждый день в диапазоне за последние 14 дней

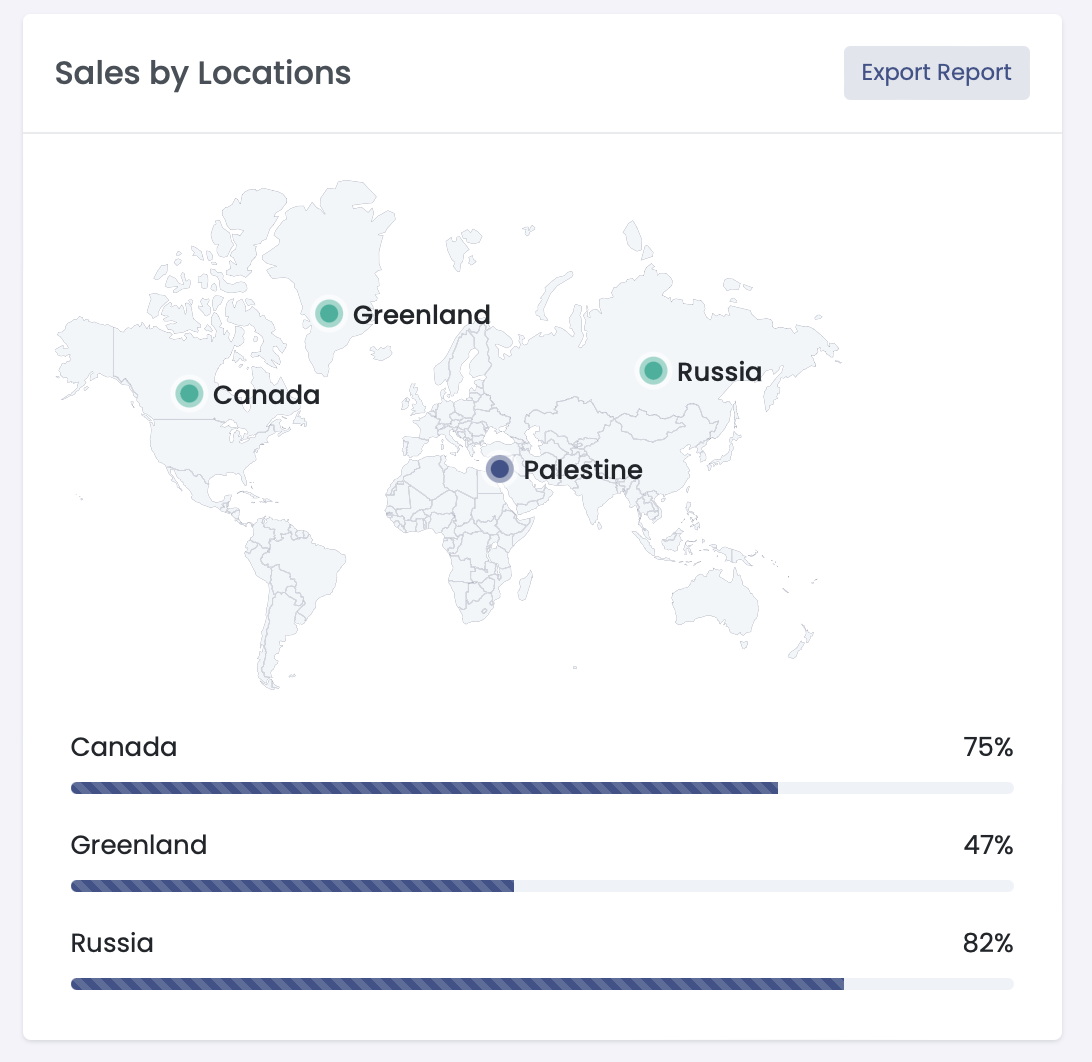
-Approved (сумма оплаченных заказов)

-Pending (сумма заказов в статусе ожидания)

-Declined (сумма заказов c неуспешной оплатой)



1. Sales by country.



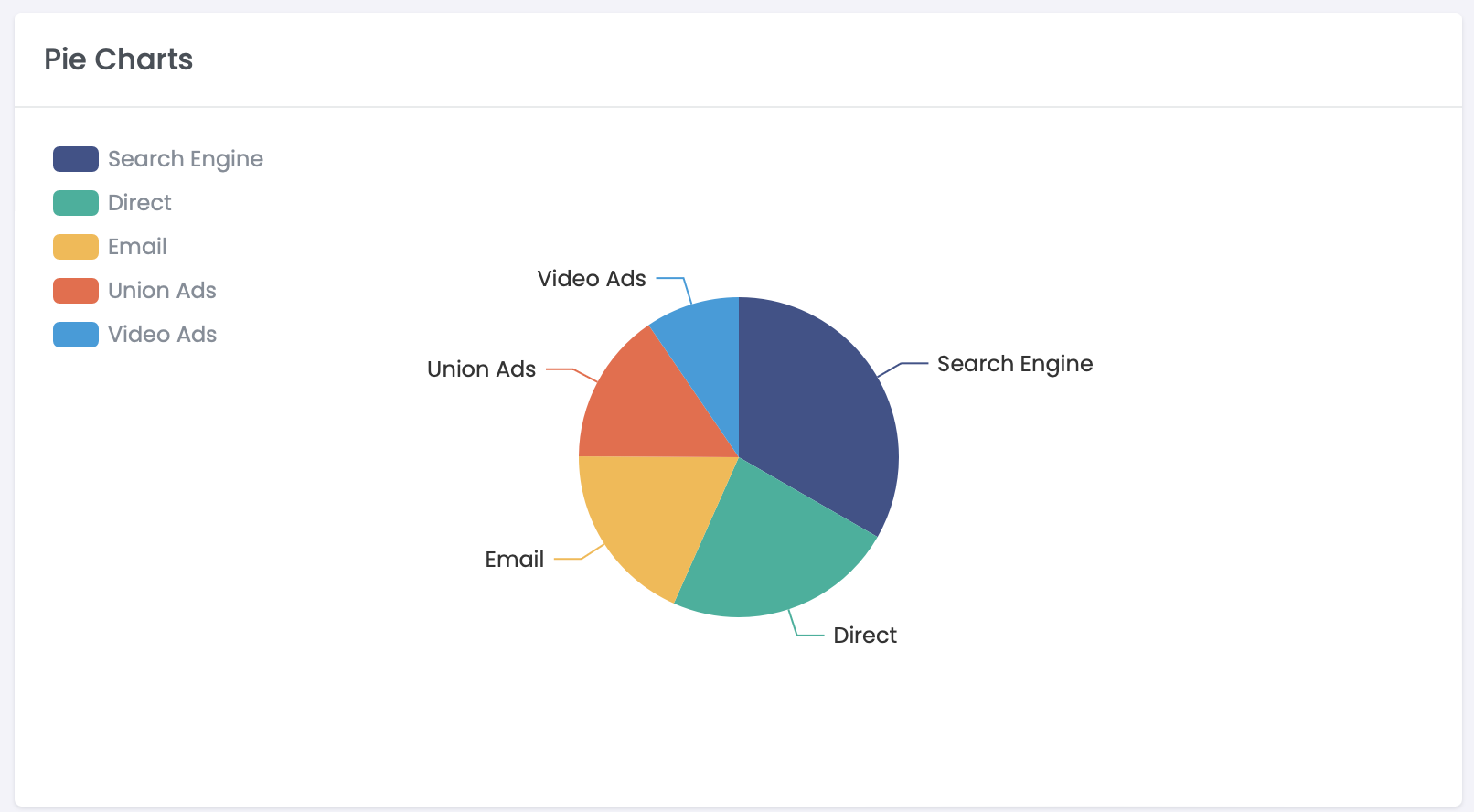
1. Revenue.

В диапазон дат добавить 3M.

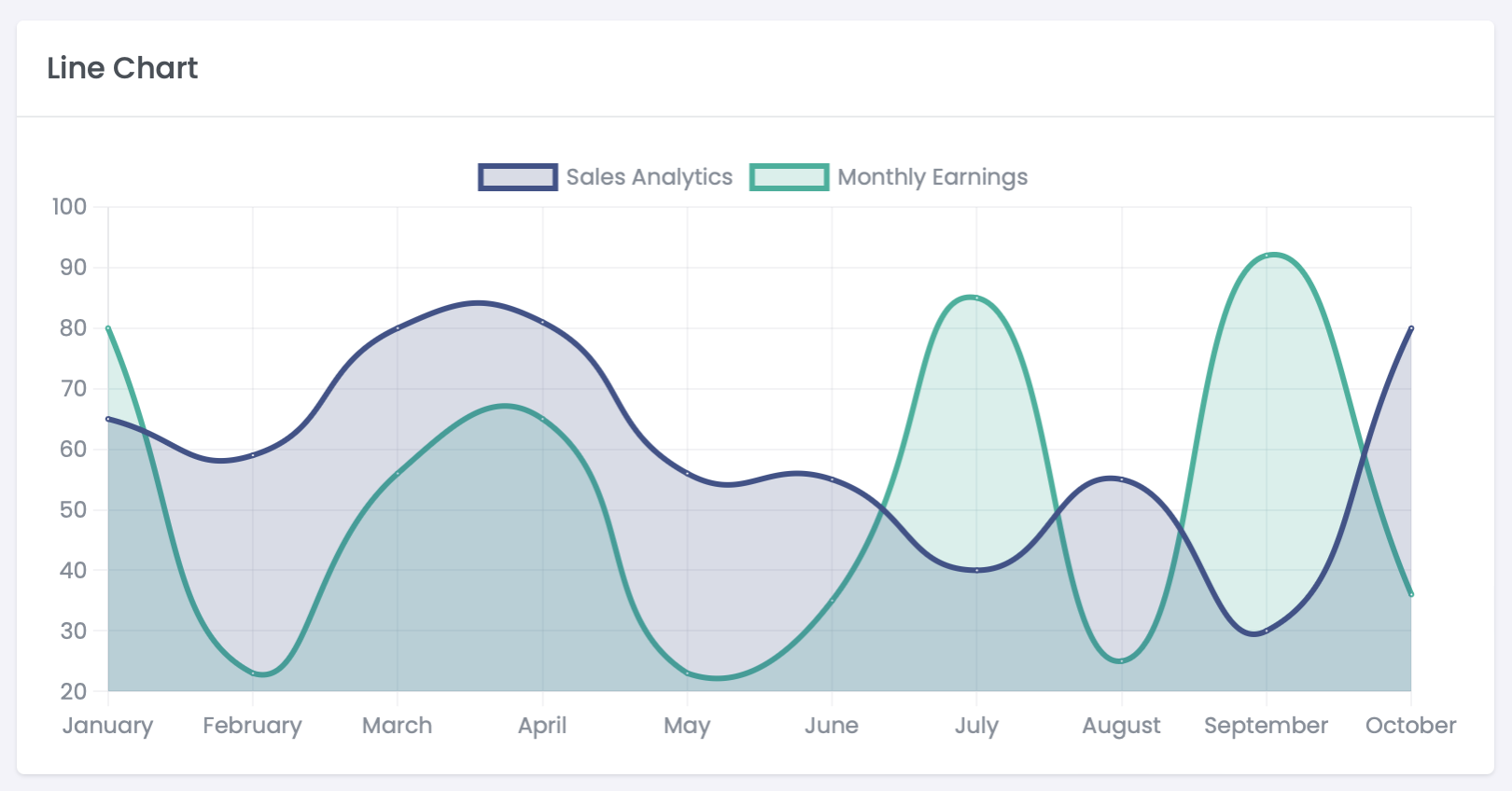
Отображать на графике Orders(количество оплаченых заказов), NET Earnings(сумма оплаченных заказов), Declines(Сумма неуспешно оплаченных заказов), Refunds(Сумма возвратов денег)



1. Device statistics. Статистика по устройствам покупателей.

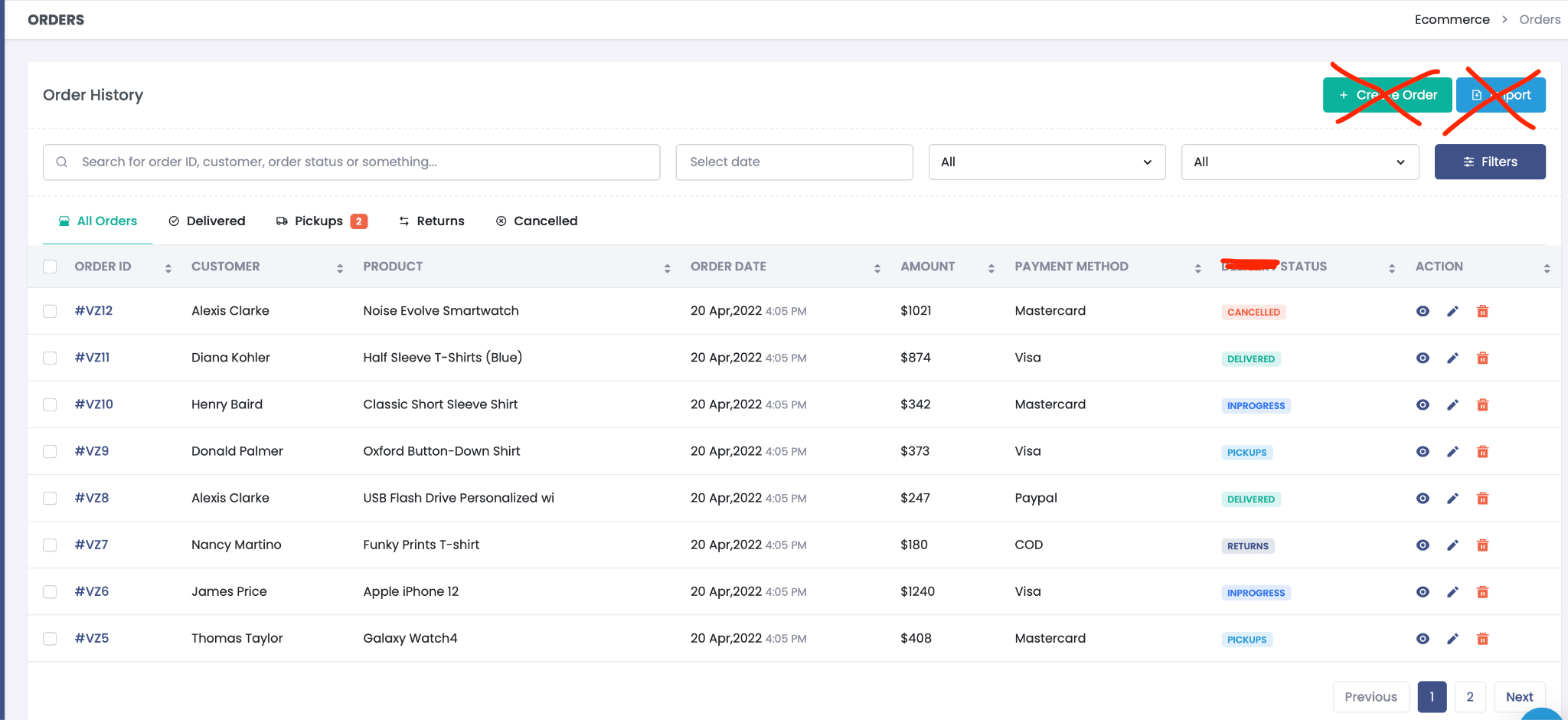


1. Monthly Earnings



—-----------------------------------------

**Orders**



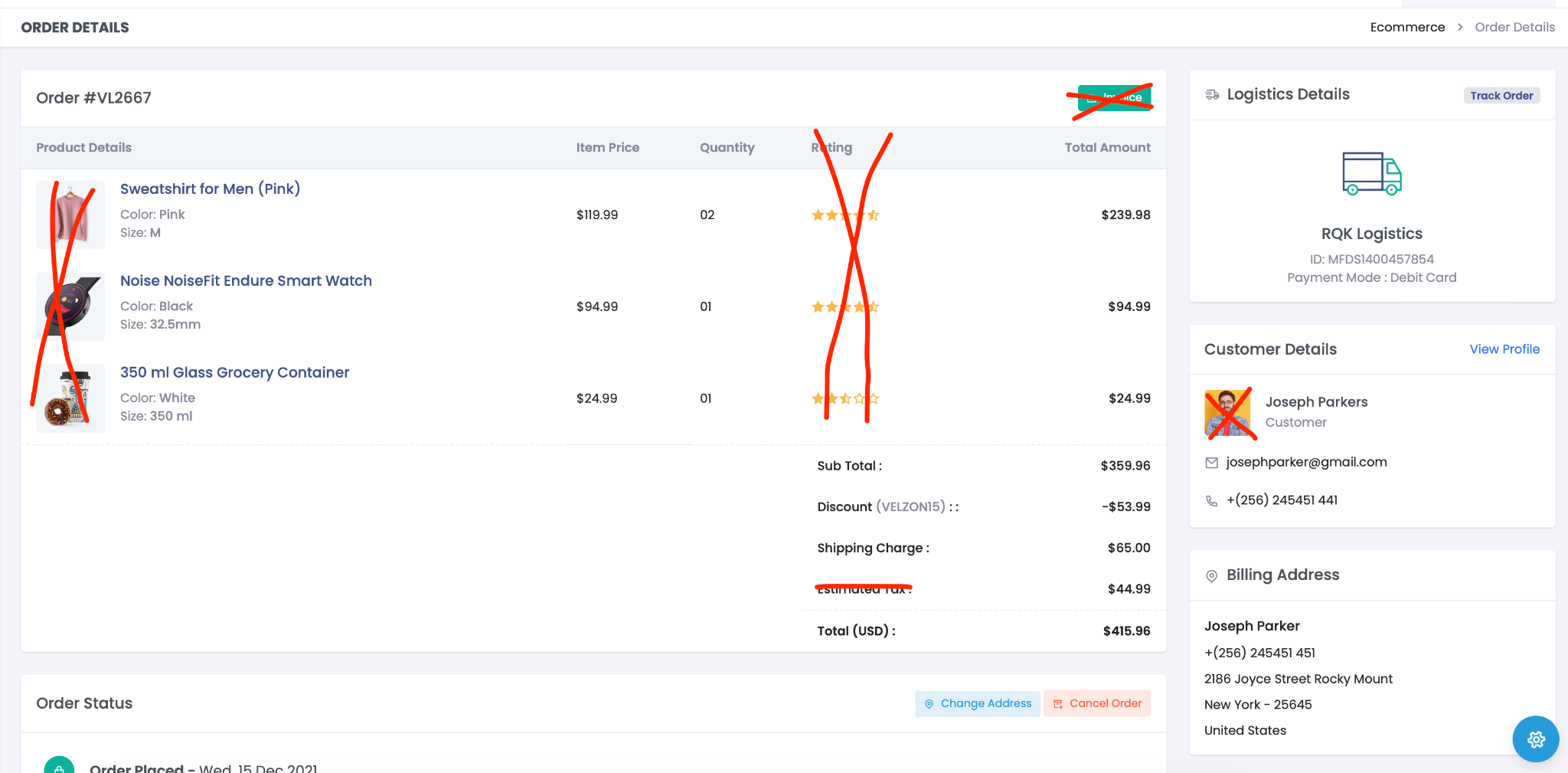
Эта страница подходит практически по всему.

1. Для групповых действий с заказами, выделенных галочками, нужно добавить кнопки ручного выставления статусов APPROVE, DECLINE, PENDING, CLOSE, REFUND, CHARGEBACK, SEND TO SHIPPER (отправить на упаковку), CALL CENTER
2. Добавить кнопку SEND TO SHIPPER для всех заказов (когда ни один заказ НЕ выделен галочкой) которые находятся в статусе успешной оплаты, но еще не были запакованы либо отправлены.

—-----------------------------------------

**Order details**

[**https://themesbrand.com/velzon/html/master/apps-ecommerce-order-details.html**](https://themesbrand.com/velzon/html/master/apps-ecommerce-order-details.html)

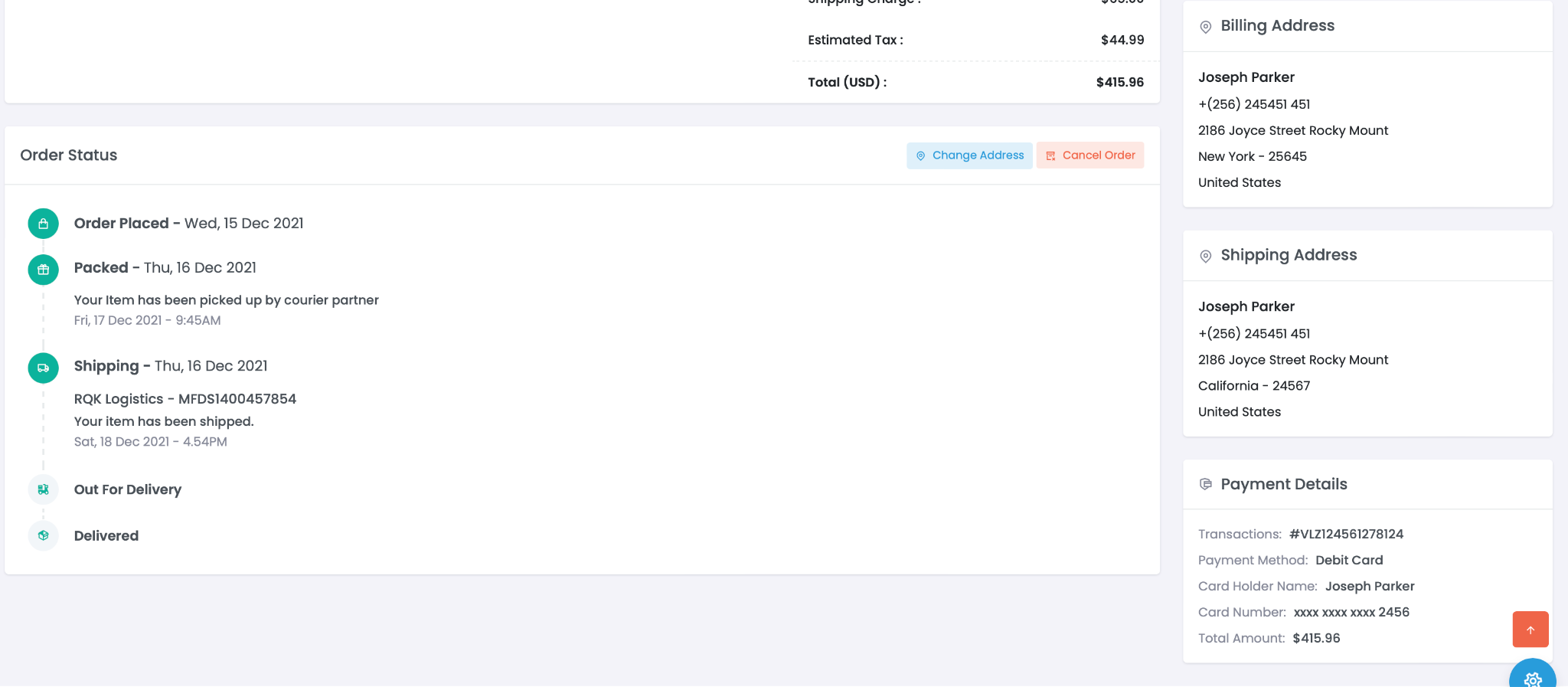
****

1. В Logistic details отображать трек номер когда он будет добавлен в систему, кнопку Track order потом доделаем. Также нужно добавить туда кнопки:

-Send to shipper

-Send tracking number

-Reship (тоже самое что Send to shipper, только для повторной отправки заказа)



1. В Customer details отображать сколько раз он сделал покупку. **Идентификация покупателей происходит по емейлу.**
2. В зону Order Status,или куда влезет добавить кнопки:

-Approve

-Decline

-Cancel

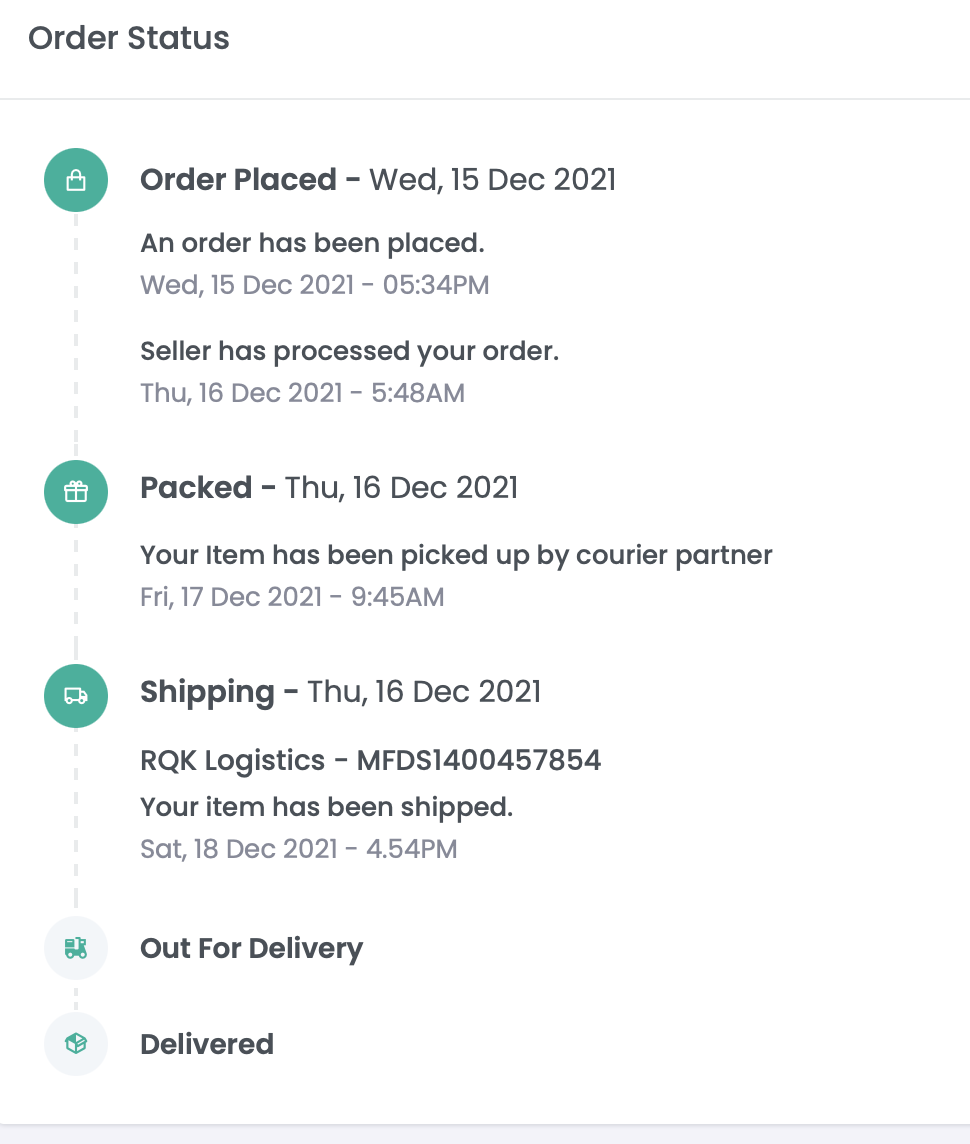
-Close

-Add to call center

-Refund

-Chargeback

1. Этот таймлайн тоже делаем чтобы были видно историю:



1. В Payment details отображать платежный метод и остальную инфу по платежу, если она есть для текущего платежного метода.

Добавить кнопку **Manual payment** для повтора платежа вручную менеджером. В зависимости от метода платежа сценарии manual payment разные. Для paypal, например это письмо со ссылкой на оплату. Для карты - это отправка платежных данных на процессинг и получение ответа. Крипта - ссылка на оплату.

1. Логирование.

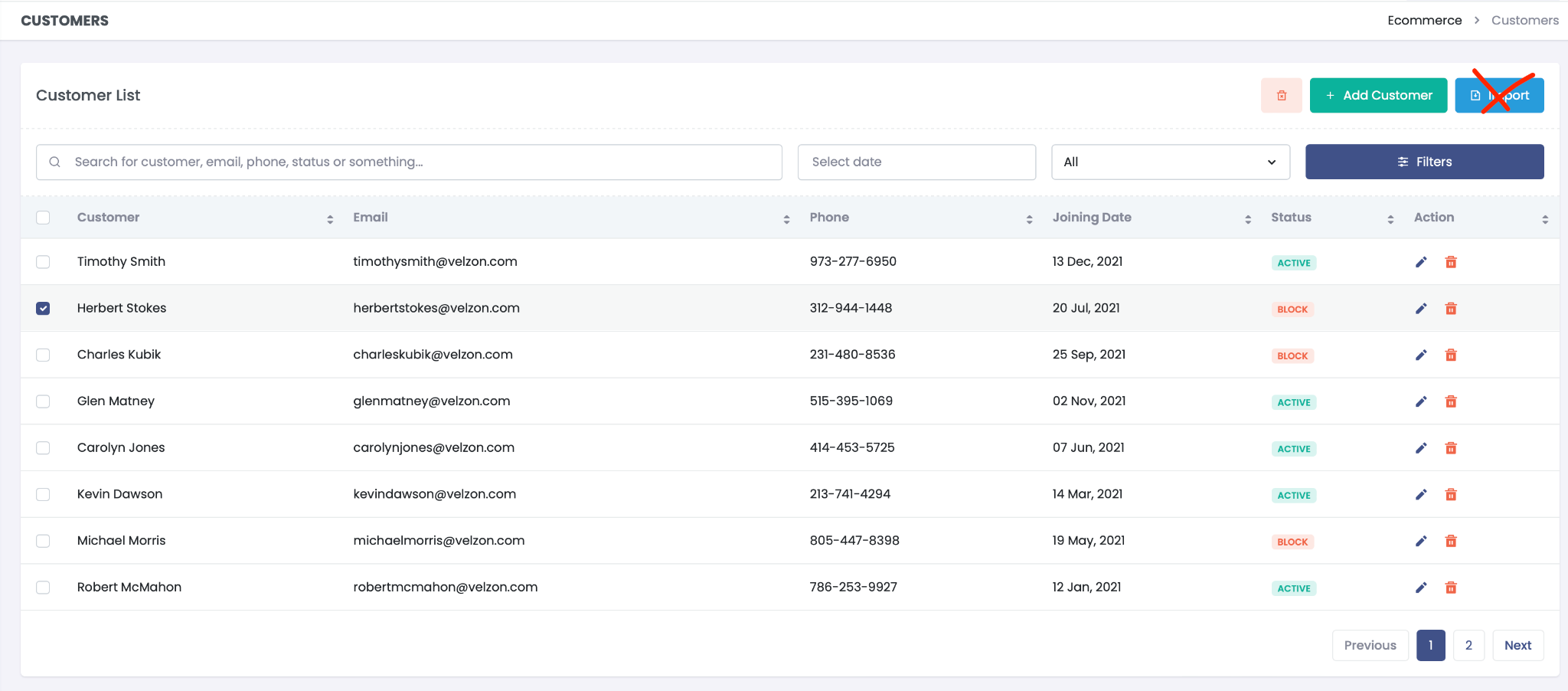
Нужно логировать все действия с заказом и выводить логи на странице Order details. Бывает такое что, например, после отправки на процессинг, оплата не происходит и ошибка не приходит, или ошибка случилась по ходу дела, а заказ так и висит в статусе пендинг и не понятно что с ним. Было бы удобно видеть что происходило с заказом, когда какой статус выставлялся, какие ошибки произошли и тп.

—-----------------------------------------

**Customers**

[**https://themesbrand.com/velzon/html/master/apps-ecommerce-customers.html#**](https://themesbrand.com/velzon/html/master/apps-ecommerce-customers.html#)

**Покупателей мы идентифицируем именно по емейлу!**



—-----------------------------------------

**Call Center**

У колл центра должен быть свой акк. В црмке им НЕ нужен dashboard.

Только страницы **Orders и Order Details.**

Сюда попадают заказы со статусом Сall Сenter, который выставляет менеджер.

Оператор не может удалять заказы из базы, либо отправлять их на повторную оплату. Но видит все что происходило с заказом. Оператор может выставлять свои статусы для ордера, то есть только те, которые касаются связи с клиентом и работы оператора колл центра:

1. New Inquiry - Заказ попал в колл центр и с ним еще ничего не делали.
2. Resolved- Запрос успешно решен.
3. Awaiting Response - Менеджер связался с клиентом и ждет от него доп. инфы.
4. Cancelled - Заказ отменен клиентом.
5. Failed - Не удалось связаться.
6. Closed - Заказ закрыт.
7. Refund Initiated - Клиент запросил возврат средств. (Меняется на Refunded когда менеджер выставил статус Refunded)
8. Reship Initiated - Клиент запросил повторную отправку посылки. (Меняется на Reshipped когда менеджер выставил статус Reshipped)

Менеджер должен видеть статусы от колл центра.