Модульные тесты:

1. Проверка card\_number регулярным выражением на соответствие шаблону — 16 цифр.

2. Проверка card\_holder регулярным выражение на соответствие шаблону — только латинские символы и дефис. Не более одного пробела.

3. Проверка card\_expiration на актуальность. Проверка, что первые две цифры не меньше 1 и не больше 12.

4.Проверка cvv регулярным выражением — 3 цифры.

5. Проверка order\_number длинны строки — от 1 до 16 символов.

6. Проверка sum на строку.

В случае любой из ошибок ответ 400.

Интеграционные тесты.

Проверка фронт-бек.

1. Если card\_expiration первые 2 цифры меньше 1 или больше 12, то на фронте выделяется поле и выводится ошибка «Проверьте скок карты».

2. Если card\_expiration вторые 2 цифры меньше последних двух цифр текущего года, то на фронте выделяется поле и выводится ошибка «Проверьте скок карты».

3. Если cvv код не соответствует шаблону, то на фронте выделяется поле и выводится ошибка «Проверьте cvv код».

Проверка бек-сервис А.

Если нет ни одной ошибок валидации, то кидаем запрос сервису.

1. Проверка на 200 ответ от сервиса при соблюдении всех правил валидации данных с фронта.

Проверка бек-репозиторий.

Если сервис вернул 200 ответ, то проверяем, что произошла запись данных в бд.

1. Проверка, что есть запись с переданным $orderNumber.

2. Проверка, что floatval(sum) == $sum).

Системные тесты.

Для удешевления на время тестирование можно ввести преобразование суммы. Скажем, в 10 раз меньше. При отправке запроса в сервис А делим sum на 10.

1. Если все данные с фронта валидны, сумма уменьшена в 10 раз, сервис А вернул 200 и репозиторий выполнил запрос, то вернуть на фронт 200 ответ с текстом «Оплата прошла успешно»

2. Если данные фронта валидны, но сервис А вернул ответ, отличный от 200, то на фронте выводится ошибка «Произошла ошибка оплаты. Повторите попытку или обратитесь в чат.»