**Домашнее задание к вебинару №23 «Unit-тестирование»**

*sum* (сумма оплаты, разделитель дробной части **запятая**, поэтому и строка, а не число).

**Задача:**

1. Модульные тесты:

Проверка валидности данных фронта:

* 1. Если *card\_holder* содержит более одного пробела, то тестируемый метод возвращает 400 с сообщением об ошибке;
  2. Если *card\_holder* содержит кириллицу, то тестируемый метод возвращает 400 с сообщением об ошибке;
  3. Если *card\_holder* содержит более одного дефиса, то тестируемый метод возвращает 400 с сообщением об ошибке;
  4. Если *card\_holder* содержит спец.символ, то тестируемый метод возвращает 400 с сообщением об ошибке;
  5. Если *card\_holder* содержит более Х символов (кол-во символов максимальное должно быть ограничено), то тестируемый метод возвращает 400 с сообщением об ошибке;
  6. Если *card\_number* содержит более 16 символов, то тестируемый метод возвращает 400 с сообщением об ошибке;
  7. Если *card\_number* содержит любой символ кроме цифр, то тестируемый метод возвращает 400 с сообщением об ошибке;
  8. Если *card\_expiration* содержит более 7 символов, то тестируемый метод возвращает 400 с сообщением об ошибке;
  9. Если *card\_expiration* содержит менее 5 символов, то тестируемый метод возвращает 400 с сообщением об ошибке;
  10. Если *card\_expiration* не содержит один /, то тестируемый метод возвращает 400 с сообщением об ошибке;
  11. Если *card\_expiration* содержит буквы, то тестируемый метод возвращает 400 с сообщением об ошибке;
  12. Если *card\_expiration* содержит спец.символы, то тестируемый метод возвращает 400 с сообщением об ошибке;
  13. Если *card\_expiration* содержит первые 2 цифры 00, то тестируемый метод возвращает 400 с сообщением об ошибке;
  14. Если *card\_expiration* содержит первые 2 цифры больше 12, то тестируемый метод возвращает 400 с сообщением об ошибке;
  15. Если 7-значный *card\_expiration* содержит последние 4 цифры больше текущий год + 20 лет, то тестируемый метод возвращает 400 с сообщением об ошибке;
  16. Если 7-значный *card\_expiration* содержит последние 4 цифры меньше текущий год - 20 лет, то тестируемый метод возвращает 400 с сообщением об ошибке;
  17. Если 5-значный *card\_expiration* содержит последние 2 цифры больше текущий год + 20 лет (аналогично берем 2 цифры от суммы), то тестируемый метод возвращает 400 с сообщением об ошибке;
  18. Если 5-значный *card\_expiration* содержит последние 2 цифры меньше текущий год - 20 лет (аналогично берем 2 цифры от суммы), то тестируемый метод возвращает 400 с сообщением об ошибке;
  19. Если *cvv* содержит более 3 символов, то тестируемый метод возвращает 400 с сообщением об ошибке;
  20. Если *cvv* содержит менее 3 символов, то тестируемый метод возвращает 400 с сообщением об ошибке;
  21. Если *cvv* содержит любой символ кроме цифр, то тестируемый метод возвращает 400 с сообщением об ошибке;
  22. Если *order\_number* содержит более 16 символов, то тестируемый метод возвращает 400 с сообщением об ошибке;
  23. Если *sum* содержит более одной запятой, то тестируемый метод возвращает 400 с сообщением об ошибке;
  24. Если sum содержит любой символ кроме, и цифр, то тестируемый метод возвращает 400 с сообщением об ошибке;
  25. Если *card\_holder, card\_number, card\_expiration, cvv, order\_number, sum* валидны (прошли требования модульных тестов выше) отправляется api запрос к сервису А

1. Интеграционные тесты:
   1. Проверяем связку «фронт-бэк»:
      1. Если *card\_holder* содержит более одного пробела, то после получения ответа от бэка на фронте выделяется поле «Номер карты»красной рамкой;
      2. Если *card\_holder* содержит более одного пробела, то после получения ответа от бэка на фронте выделяется поле «Номер карты»красной рамкой;
      3. Если *card\_holder* содержит кириллицу, то после получения ответа от бэка на фронте выделяется поле «Номер карты»красной рамкой;
      4. Если *card\_holder* содержит более одного дефиса, то после получения ответа от бэка на фронте выделяется поле «Номер карты»красной рамкой;
      5. Если *card\_holder* содержит спец.символ, то после получения ответа от бэка на фронте выделяется поле «Номер карты»красной рамкой;
      6. Если *card\_holder* содержит более Х символов (кол-во символов максимальное должно быть ограничено), то после получения ответа от бэка на фронте выделяется поле «Номер карты»красной рамкой;

Если *card\_number* содержит более 16 символов, то после получения ответа от бэка на фронте выделяется поле «Номер карты»красной рамкой;

Если *card\_number* содержит любой символ кроме цифр, то после получения ответа от бэка на фронте выделяется поле «Номер карты»красной рамкой;

Если *card\_expiration* содержит более 7 символов, то после получения ответа от бэка на фронте выделяется поле «Номер срока карты»красной рамкой;

Если *card\_expiration* содержит менее 5 символов, то после получения ответа от бэка на фронте выделяется поле «Номер срока карты»красной рамкой;

Если *card\_expiration* не содержит один /, то после получения ответа от бэка на фронте выделяется поле «Номер срока карты»красной рамкой;

Если *card\_expiration* содержит буквы, то после получения ответа от бэка на фронте выделяется поле «Номер срока карты»красной рамкой;

Если *card\_expiration* содержит спец.символы, то после получения ответа от бэка на фронте выделяется поле «Номер срока карты»красной рамкой;

Если *card\_expiration* содержит первые 2 цифры 00, то после получения ответа от бэка на фронте выделяется поле «Номер срока карты»красной рамкой;

Если *card\_expiration* содержит первые 2 цифры больше 12, то после получения ответа от бэка на фронте выделяется поле «Номер срока карты»красной рамкой;

Если 7-значный *card\_expiration* содержит последние 4 цифры больше текущий год + 20 лет, то после получения ответа от бэка на фронте выделяется поле «Номер срока карты»красной рамкой;

Если 7-значный *card\_expiration* содержит последние 4 цифры меньше текущий год - 20 лет, то после получения ответа от бэка на фронте выделяется поле «Номер срока карты»красной рамкой;

Если 5-значный *card\_expiration* содержит последние 2 цифры больше текущий год + 20 лет (аналогично берем 2 цифры от суммы), то после получения ответа от бэка на фронте выделяется поле «Номер срока карты»красной рамкой;

Если 5-значный *card\_expiration* содержит последние 2 цифры меньше текущий год - 20 лет (аналогично берем 2 цифры от суммы), то после получения ответа от бэка на фронте выделяется поле «Номер срока карты»красной рамкой;

Если *cvv* содержит более 3 символов, то после получения ответа от бэка на фронте выделяется поле «cvv»красной рамкой;

Если *cvv* содержит менее 3 символов, то после получения ответа от бэка на фронте выделяется поле «cvv»красной рамкой;

Если *cvv* содержит любой символ кроме цифр, то после получения ответа от бэка на фронте выделяется поле «cvv»красной рамкой;

Если *order\_number* содержит более 16 символов, то после получения ответа от бэка на фронте выделяется поле «cvv»красной рамкой;

Если *sum* содержит более одной запятой, то после получения ответа от бэка на фронте выводится модальное окно с текстом “ошибка оформления заказа”

Если sum содержит любой символ кроме, и цифр, то после получения ответа от бэка на фронте выводится модальное окно с текстом “ошибка оформления заказа”

* 1. Проверяем связку «бэк-репозиторий»:
     + 1. Если вызывать метод записи в БД, то в репозиторий добавляется новая запись
  2. Проверяем связку «бэк-сервис оплаты А»:
     + 1. Если из *card\_holder, card\_number, card\_expiration, cvv, order\_number, sum НЕ* валидно хотя б одно поле (НЕ прошло требования модульных тестов) НЕ отправляется api запрос к сервису А

1. Системные тесты:
   1. Если *cvv* неверен (но валиден, как и остальные поля при оплате по требованиям модальных тестов), то после получения ответа от бэка (ответ сервиса А):

* на фронте выводится сообщение об ошибке «Не удалось списать данные с карты, проверьте реквизиты»
* в БД запись об оплате не добавляется
* в списке ЗК в личном кабинете долг равен сумме заказа
  1. Если данные карты валидны (и данные тестовой карты), то после получения ответа от бэка:
* на фронте выводится сообщение «Заказ успешно оплачен».
* в БД добавляется запись об оплате Заказа клиента.
* в списке ЗК в личном кабинете долг равен 0.
  1. Если данные карты НЕ валидны (любое хотя б одно поле), то после получения ответа от бэка:
* на фронте не валидное поле получает красный цвет
* в БД НЕ добавляется запись об оплате Заказа клиента.
* в списке ЗК в личном кабинете долг равен сумме ЗК.

Варианты удешевления:

1. Поскольку мы тестировали на интеграционном уровне, что api запрос не уходит к платежному сервису при НЕ валидных данных хотя б одного поля, тест с системного уровня не имеет смысла (можно не писать)