1. Модульные тесты:

|  |  |  |
| --- | --- | --- |
| **Тест-кейс: Не заполнено поле формы** | | |
| **Проверка поля формы:** | **Код ответа:** | **Результат теста:** |
| card\_number |  | assertEmpty($card\_number, "Поле номер карты не заполнено"); |
| card\_holder | assertEmpty($card\_number, “Поле держатель карты не заполнено”); |
| card\_expiration | assertEmpty($card\_number, “Поле окончание действия карты не заполнено” |
| cvv | assertEmpty($card\_number, “Поле код с обратной стороны карты не заполнено” |
| order\_number | assertEmpty($card\_number, “Поле номер заказа не заполнено” |
| sum | assertEmpty($card\_number, “Поле сумма не заполнено”); |

|  |  |  |  |
| --- | --- | --- | --- |
| **Тест-кейс: Данные введены неверно** | | | |
| **Проверка поля формы:** | **Условие проверки:** | **Код ответа:** | **Результат теста:** |
| card\_number | Количество цифр не равно 16 |  | assertRegExp(‘/^[0-9]{16}$/’,$ card\_number, "Данные о номере карты введены не верно"); |
| card\_holder | Помимо латинских букв и пробела есть другие символы | assertRegExp(‘/[^A-Za-z-]/’,$card\_holder, “Имя и Фамилия некорректны”) ; |
| card\_expiration | Прошедшая дата, неправильный формат. Помимо цифр и разделителя / есть другие символы | assertRegExp(‘/^0[1-9]|1[012]\/\d{2}$/’, $cvv, “Данные о дате окончания действия карты введены неверно”) ; |
| cvv | Количество цифр не равно 3 | assertRegExp(‘/^[0-9][0-9][0-9]/’, $cvv, “Код состоит из 3 цифр”) ; |
| order\_number | Количество произвольных символов не больше 16 | assertTrue(false, “Данные заказа введены неверно” ); |
| sum | Помимо цифр и запятой есть другие символы | assertRegExp(‘/^[0-9]{1,8}\,[0-9]{2}/’, $sum, “Для разделителя используйте запятую”) ; |

|  |  |  |
| --- | --- | --- |
| **Тест-кейс: Данные верны** | | |
| **Проверка всех полей формы:** | **Код ответа:** | **Результат теста:** |
| card\_number, card\_holder, card\_expiration, cvv, order\_number, sum | header("HTTP/1.1 200 OK"); | Передаем данные в API-запросе на сервис A. Получаем ответ об успешной оплате. |

1. Интеграционные тесты:

**Тест-кейс: Связка «фронт-бэк»**

При результатах модульных тестов на код ответа 400, поле формы, не прошедшее валидацию, выделяется красным цветом с текстом ошибки.

|  |  |  |
| --- | --- | --- |
| **Тест-кейс: Попытка списания денег** | | |
| **Проверка всех полей формы:** | **Код ответа:** | **Результат теста:** |
| Все поля валидны | header("HTTP/1.1 200 OK"); | Сервису А удается списать деньги. Далее будет запись в БД |
| Поле cvv c ошибкой | header("HTTP/1.1 403 Списание не удалось"); | Сервису А не удается списать деньги. Возвращаем на фронт ошибку о неверном cvv |

**Тест-кейс: Связка «Бэк и Репозиторий»**

|  |  |  |
| --- | --- | --- |
| **Тест-кейс: Попытка списания денег** | | |
| **Условие проверки:** | **Код ответа:** | **Результат теста:** |
| Списание прошло успешно. Запись в БД | header("HTTP/1.1 200 OK"); | Номер заказа и сумма соответствуют записи в БД |
| При списании возникли ошибки | header("HTTP/1.1 403 Списание не удалось"); | Номер заказа и сумма отсутствуют в БД. |

с) Системные тесты:

|  |  |
| --- | --- |
| **Тест-кейс: Отправка формы** | |
| **Отправка формы:** | **Результат теста:** |
| Отправка пустой формы | Сообщение на фронте “Введите реквизиты”, все поля выделены красным |
| Отправка валидной формы | Сообщение на фронте “Оплата заказа прошла успешно” |
| Отправка формы с ошибками | Сообщение на фронте “Введите данные корректно”, поля с ошибками выделены поля красным цветом |
| Отправка валидной формы, но не с существующими данными | Сообщение на фронте “Держатель карты (и)или реквизиты указаны неверно. Отмена транзации” |