|  |  |
| --- | --- |
| Название проекта | Обработка E-commerce транзакций |
| Название продукта | Эквайринг e-commerce транзакции |

План тестирования

Версия документа: 0.1

Дата: 23.04.2020

Подготовлено Пушкиным Никитой

Вступление

Данный документ описывает план тестирования функциональности продукта “Эквайринг e-commerce транзакции”. Продукт состоит из таких компонентов, как страница приема платежных данных, страница просмотра статуса страница для просмотра прошедших транзакций, база данных и серверное приложения, обрабатывающее все входящие и выходящие запросы.

Данный план тестирует взаимодействие различных компонентов приложения друг с другом. Должна быть протестирована совместная работа страницы приема платежных данных и серверного приложения, серверного приложения и базы данных, страницы просмотра прошедших транзакций и серверного приложения, а также серверного приложения и банка-эмитента (имитируемого Java кодом при помощи фреймворка Mockito).

Ссылки

Здесь будут ссылки на другие документы. План базы данных, проектная документация, мб что-то еще.

Тестируемые компоненты

1. Страница ввода суммы и страница приема платежных данных;
2. Страница для просмотра прошедших транзакций;
3. База данных;
4. Серверное приложения.

Тестируемая функциональность

1. Все формы для отправки данных должны корректно отображаться и взаимодействовать с приложением.
2. На странице /invoice при вводе суммы должен создаваться соответствующий InvoicePostView, по нему строится InvoiceView, а пользователь перенаправляется на страницу ввода информации о карте. По одному заказу может быть создано несколько InvoicePostView.
3. На странице /invoice/{id} пользователь вводит информацию о карте, после чего должна быть проведена внутренняя валидация карты и отправка данных о заказе и карте в банк-эмитент, а транзакция должна быть сохранена в базе данных. Должна быть возможность просматривать информацию о статусе транзакции (get-запрос).
4. На странице '/transaction/{transactionId}' по методу put изменяется ее статус.
5. На странице ‘/transactions’ перечислены все прошедшие через систему транзакции.

Подход

Тестирование будет проводиться непосредственным испытанием функциональности, описанной выше. Сначала будет сымитировано взаимодействие фронтенда и бэкенда, а затем все то же самое будет протестировано с использованием интерфейса фронтенда. Взаимодействие с банком будет эмулироваться java-кодом.

Misc

Оставшиеся пункты, кажется, к нам не сильно относятся.

Сценарии тестирования

1. Пользователь вводит сумму перевода, после чего перенаправляется на страницу ввода информации о карте, а в системе создается новый invoice.
2. Пользователь вводит информацию о карте, после чего создается соответствующая транзакция, отправляющаяся на валидацию в банк, а сам пользователь перенаправляется на страницу с информацией о статусе данной транзакции.
3. Приложение делает запрос к банку на валидацию карты, после ответа от банка код ответа должен сохраняться в базе данных.
4. Пользователь по get-запросу на странице /invoice/{invoiceId} получает данные о соответствующей транзакции.
5. Пользователь на странице /transactions просматривает информацию о всех прошедших через систему транзакциях.
6. Пользователь поочередно выполняет действия, описанные в пунктах 1, 2, 4, 5.