



Integration Guide

XSOLLA PAY STATION

SDK and API Reference

Legal Notice: This document contains privileged and/or confidential information and may not be disclosed, distributed or reproduced without the prior written permission of Xsolla®.

CONFIDENTIAL | Effective Date: 10-Jul-2020



СОДЕРЖАНИЕ

1	ВСТУПЛЕНИЕ	3
1.1	ЦЕЛЬ	3
1.2	ЭКСПЛУАТАЦИОННОЕ НАЗНАЧЕНИЕ	3
1.3	ЦЕЛЕВАЯ АУДИТОРИЯ.....	3
2	ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ	4
3	ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ	5
4	ДАННЫЕ И СПИСКИ	7
4.1	ОБЩАЯ СХЕМА РАБОТЫ	7
4.1.1	Инициализирующий запрос	8
4.1.2	Выбор покупки	9
4.1.2.1	Покупка за реальную валюту	10
4.1.2.2	Покупка за виртуальную валюту	10
5	PURCHASING USER FLOW	13



1 ВСТУПЛЕНИЕ

1.1 ЦЕЛЬ

Мы создали этот SDK, чтобы разработчикам альтернативных платформ было проще использовать платежные решения Xsolla. SDK снижает порог вхождения при использовании продукта на конкретной платформе. Одно из заданий этого документа — помочь в написании библиотеки, реализующей функциональность магазина на основе Pay Station API.

You can use this SDK to implement the Xsolla payment solution on multiple computing platforms. The SDK reduces the learning curve when you use the product with a specific platform. Use this guide to write a library for the store that uses Pay Station API.

1.2 ЭКСПЛУАТАЦИОННОЕ НАЗНАЧЕНИЕ

Цель создания данного продукта:

- Предоставить удобный инструмент для разработчиков, который бы снизил трудозатраты по внедрению биллингового решения компании Xsolla.
- Улучшить user experience конечного пользователя: инструмент должен работать на всех платформах, требовать меньше дополнительных переходов в другие окна (или исключить такие переходы).

We created this product to:

- Provide a convenient tool for developers to reduce man-hours on implementing the Xsolla billing solution.
- Improve end user experience: the tool should be platform agnostic, require fewer clicks to go to other windows, or stay within the same window.

1.3 ЦЕЛЕВАЯ АУДИТОРИЯ

Этот документ предназначен для таких групп пользователей:

- Разработчики, которые внедряют решение Pay Station.
- Конечные пользователи, которые используют решение.

This guide is intended for these groups of users:

- Developers who implement the Pay Station solution.
- End users who use the solution.



2 ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

- **Header** — верхняя часть окна магазина; включает в себя все до пунктов навигационного меню включительно.
- **Footer** — нижняя часть окна магазина.
- **Серверная интеграция** — интеграция и взаимодействие с Xsolla происходит посредством специально настроенного сервера.
- **Токен** — специальная зашифрованная строка, необходимая для взаимодействия с Xsolla.
- **Упрощенная интеграция** — сервер игры отсутствует, все взаимодействия через клиент.



3 ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

В общем виде должна быть создана библиотека, при использовании которой у разработчика не возникнет вопросов о ее назначении. Разработчику должно быть достаточно использовать одну функцию, чтобы получить полную функциональность электронного магазина с модулями виртуальной валюты, виртуальных товаров, подписок и поддержкой множества платежных методов.

Посредством взаимодействия с API должна быть реализована отрисовка магазина со всеми внутренними элементами интерфейса, включающая в себя такие экраны:

- Экран виртуальных товаров
- Экран виртуальной валюты
- Экран подписок
- Экран выбора платежных методов
- Экран оплаты
- Экран статуса
- Экран ошибки

Необходимо реализовать два варианта взаимодействия с [Pay Station API](#): в формате запросов POST или GET.

1. Серверная интеграция — передается строка токена.
2. Упрощенная интеграция — передается JSON.

```
{
  "user":{
    "id":{
      "value":0
    },
    "name":"",
    "email":"",
    "country":{
      "value":"",
      "allow_modify":false
    }
  },
  "settings":{
    "project_id":0,
    "language":"",
    "currency":"",
    "mode":"sandbox",
    "secretKey":""
  }
}
```

Для варианта с JSON:

- Подготовить специальный параметризованный объект, чтобы у разработчика не возникла необходимость самостоятельно его генерировать.
- Предоставить возможность взаимодействия с Xsolla API в режиме «песочницы»: в URL-адресе перед secure добавлять “sandbox-”. Соответственно, пользователь должен иметь возможность определить, в каком режиме проводятся платежи.
- Ожидаемый результат: функция либо объект с параметрами.



```
CreateShop(  
    string, // token or json  
    bool, // isSandbox  
    OkCallback, // возможность обработать успешный платеж  
    ErrorCallbck // возможность обработать платеж с ошибкой  
)
```

`OkCallback`, `ErrorCallbck` могут быть реализованы любым удобным способом и не обязательно должны передаваться как параметры.

После вызова данной функции управление процессом работы приложения со стороны разработчика-пользователя заканчивается и переходит под управление библиотеки. Контроль над работой приложения разработчик получает вновь лишь в момент успешного или неуспешного завершения платежа.

Порядок запросов и переходов описан в разделе [Общая схема работы](#). Построение интерфейса и обработка запросов описаны в разделе [Обработка запросов](#). Описание дизайна находится в разделе [Дизайн](#).



4 ДАННЫЕ И СПИСКИ

4.1 ОБЩАЯ СХЕМА РАБОТЫ

Общая схема взаимодействия (Рис. 1) с Pay Station API включает в себя такие шаги:

1. Инициализирующий запрос.
2. Выбор покупки.
3. Выбор платежной системы.
4. Проведение платежа.
5. Статус покупки.

Часть из этих шагов (2-3) не являются обязательными и при необходимости могут быть пропущены. Вся информация, необходимая для определения схемы, может быть получена из первого шага или инициализирующего запроса. Далее рассмотрим каждый из этих шагов в отдельности. Описание будет состоять из текстового описания и диаграммам, наглядно иллюстрирующих процесс работы сервиса.

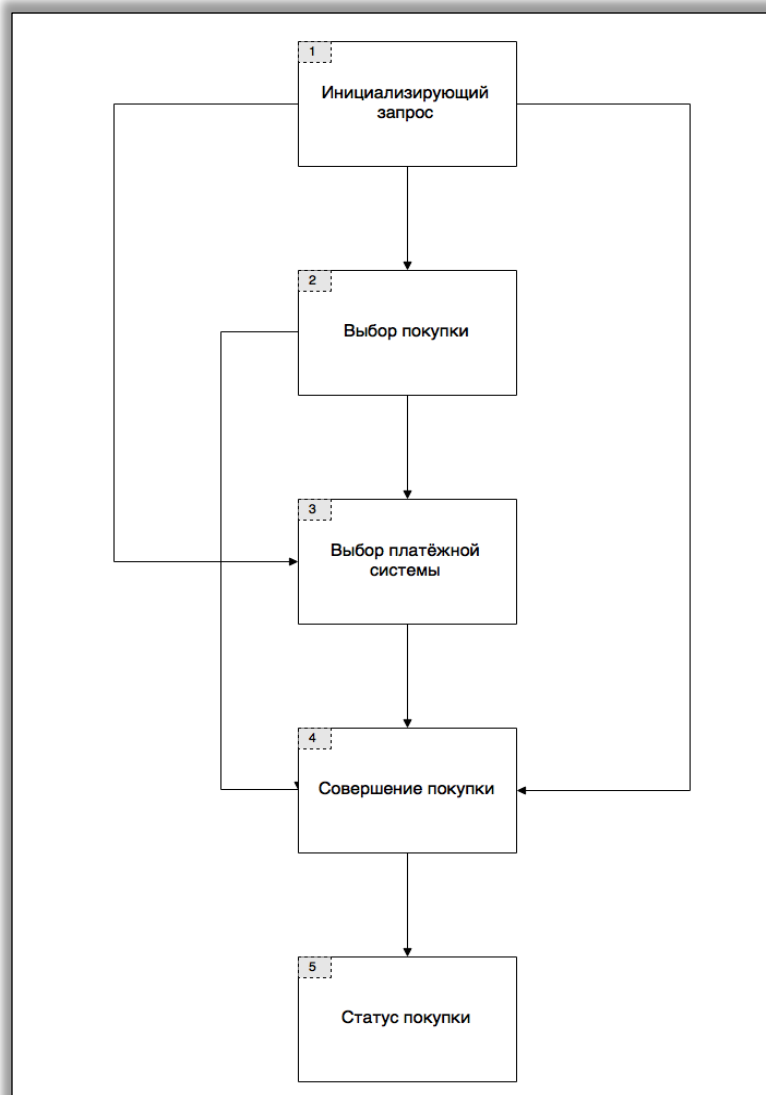


Рис. 1 Общая схема работы с Pay Station API



4.1.1 Инициализирующий запрос

Инициализирующим запросом в данном случае является запрос [Utils](#). В нем мы получаем всю необходимую информацию для работы сервиса. Ответ включает такую информацию:

- [Информация о пользователе](#)
- [Информация о проекте](#)
- [Информация о покупке](#)
- [Настройки](#)
- [Переводы](#)

Если мы получили корректный ответ от сервера, то в первую очередь нас интересует [Информация о покупке](#). С ее помощью можно определить передана ли нам покупка ("purchase"). Если да, то необходимо проверить наличие товара ("virtual_currency", "virtual_items" или "subscription") и наличие платежной системы ("payment_system").

Это необходимо для определения дальнейших наших шагов: если в ответе содержится товар, то пропустите шаг 2 (Рис. 1). Если же содержится платежная система, пропустите шаг 3 (Рис. 1). Схема запроса представлена на Рис. 2.

Здесь уже начинается построение интерфейса: на этом этапе у нас есть все необходимые данные для построения header и footer окна магазина.

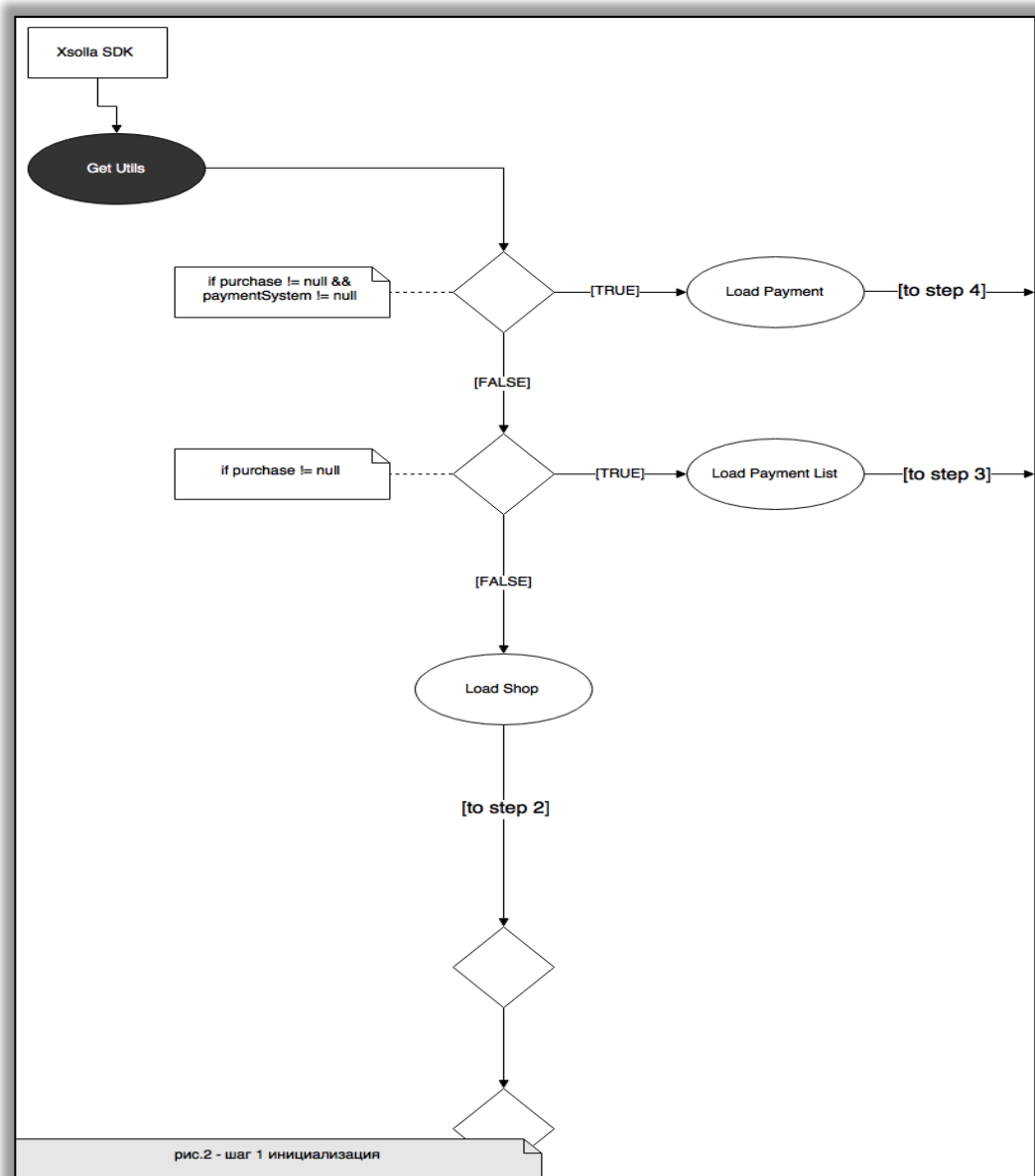


Рис. 2 Инициализация

4.1.2 Выбор покупки

Выбор покупки — это наш второй шаг, если перед этим мы не получили товар ("virtual_currency", "virtual_items" или "subscription"). Здесь нам предстоит построить магазин, в котором пользователь сможет выбрать понравившийся ему товар и продолжить покупки.

В первую очередь мы по-прежнему используем информацию из [инициализирующего запроса](#). Нас интересуют поля из объекта [настроек](#) ("settings"): "goods_at_first", "pricepoints_at_first", "subscriptions_at_first". Их значения могут быть 0 или 1. В зависимости от того, что нам пришло, мы будем отображать те или иные пункты меню [Товары](#), [Вирт валюта](#) и Подписки соответственно. В зависимости от того, что нам пришло, необходимо загрузить одну из категорий и отобразить ее для пользователя.

Та же логика работает и при переключении пунктов меню. Далее пользователь выберет какой-либо из товаров. Есть два варианта: **покупка за реальную валюту** или **покупка за виртуальную валюту**.



4.1.2.1 Покупка за реальную валюту

В случае покупки за реальную валюту мы сразу переходим к проверке пользовательского баланса. Если баланса достаточно, оплата продолжается через Xsolla Balance. В ином случае происходит открытие списка платежных систем.

4.1.2.2 Покупка за виртуальную валюту

При покупке за виртуальную валюту схема работы немного отличается. Сначала выполняется запрос Sumtagu, в котором мы получаем содержимое нашей покупки и метку, которая говорит нам, можно ли пропустить подтверждение от пользователя. Если да, то переходим к следующему запросу Proceed, который может вернуть или не вернуть ошибку. В этом случае мы продолжаем платеж опять же через Xsolla Balance. Если вернулась ошибка или необходимо подтверждение от пользователя, мы попадаем на экран подтверждения платежа. На этом экране пользователю показана его покупка, выбор варианта «Спрашивать ли подтверждение в следующий раз?» и ошибка, если мы вернулись сюда после запроса Proceed.

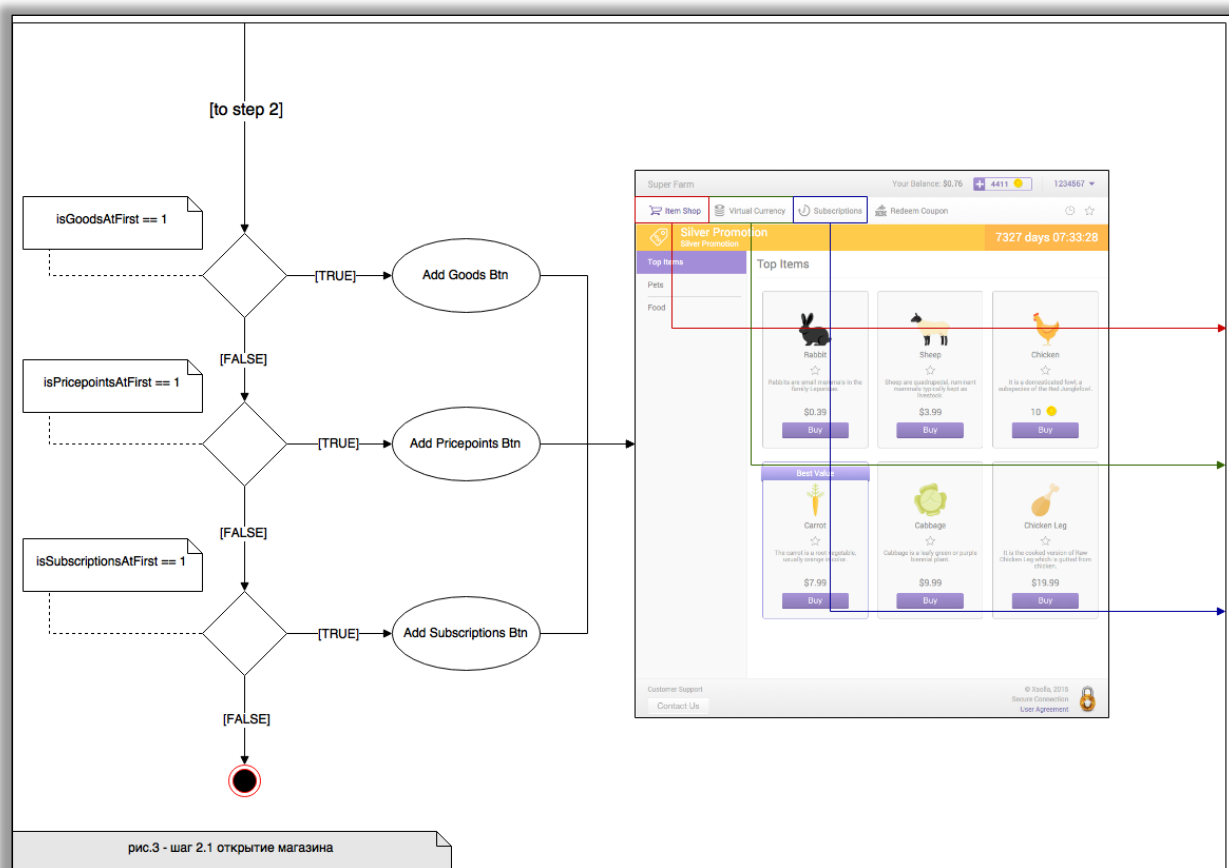


Рис. 3 Открытие магазина

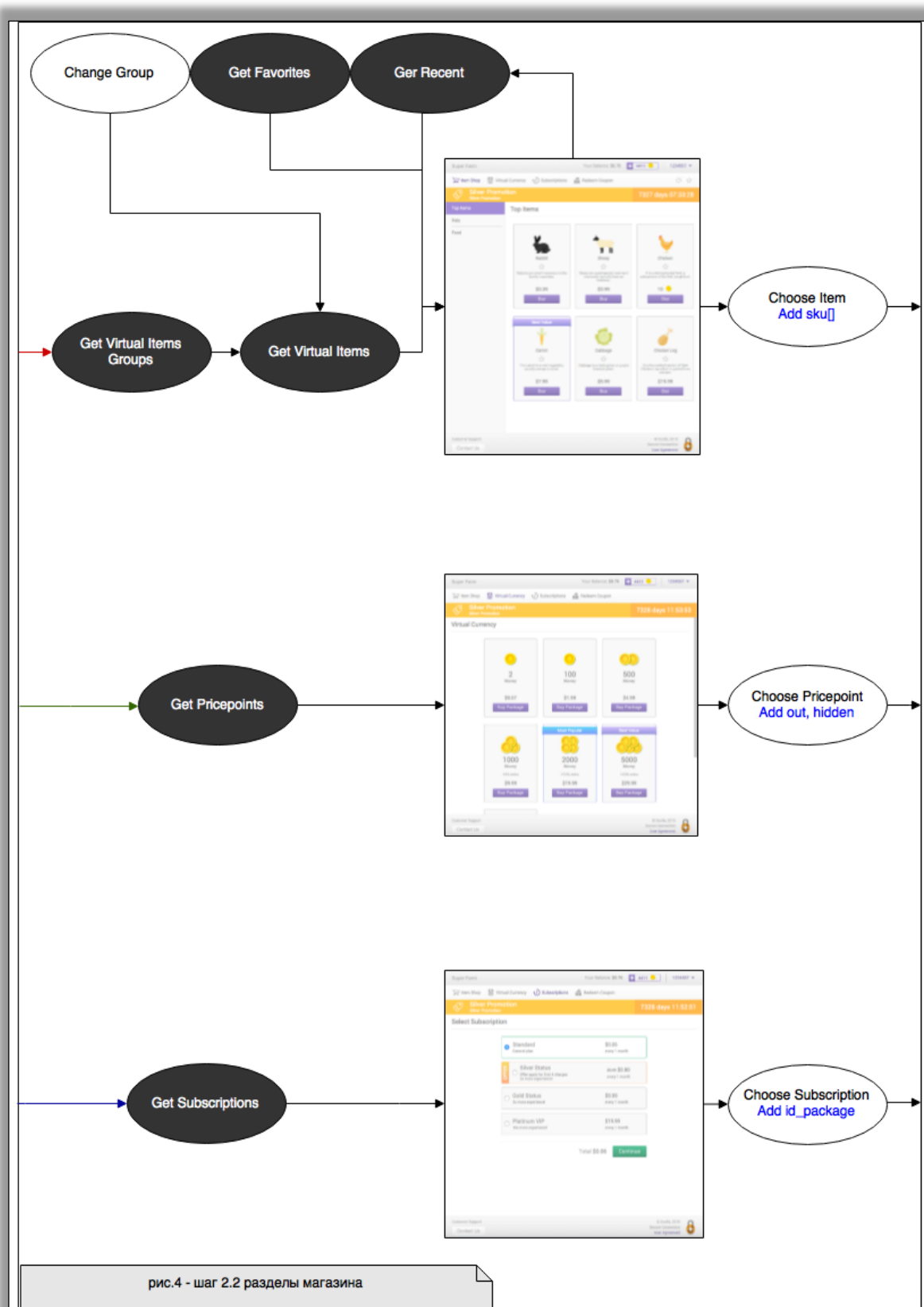


Рис. 4 Магазин — навигация

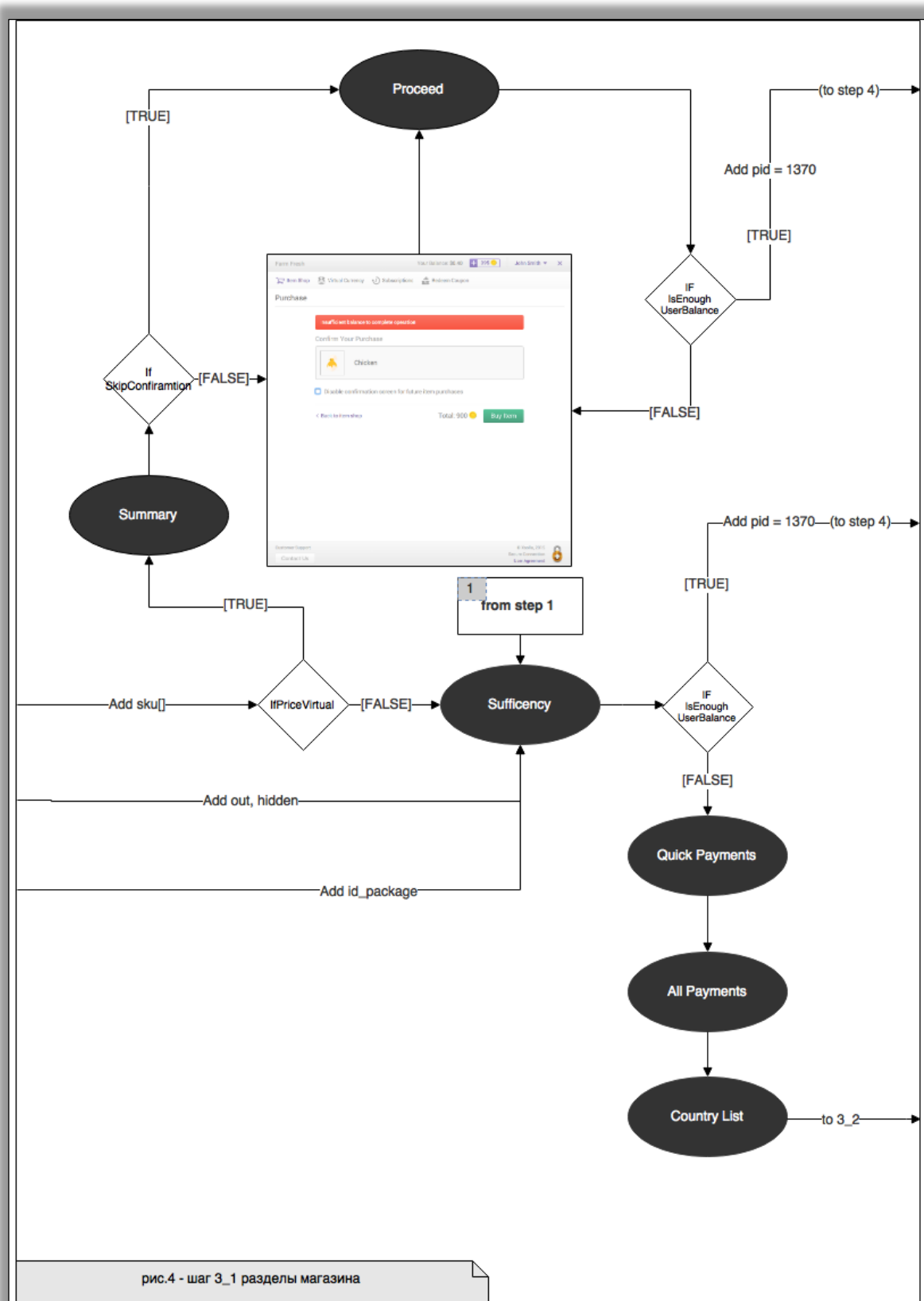


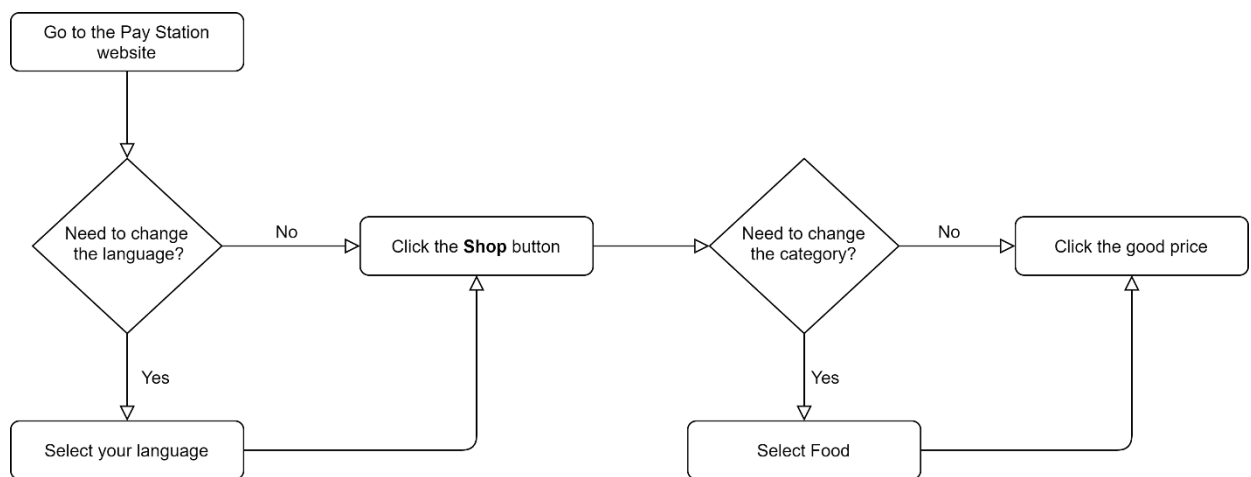
Рис. 5 Магазин — переход к выбору платежки



5 PURCHASING USER FLOW

The user flow below shows how you can buy virtual goods on the Pay Station demo website. This flow ends when you have to enter your credit card information.

1. Go to the [Pay Station](#) website.
2. Select your language. English is the default language.
3. Click the **Shop** button.
4. Select the category for the goods you wish to buy:
 - Pets
 - Food
5. Scroll-down to view the list of all goods for the selected category.
6. To buy the good, click its price.





История версий					
Версия	Краткое описание внесенных изменений (причина изменений)	Кем пересмотрено (автор изменений)	Дата	Согласования	
				Утвердили (ФИО)	Дата
0.1	Вычитка исходной версии документа	Иван Чебан	10-Jul-2020		