

1 Общие сведения

SDK создаётся для упрощения использования платежного решения Xsolla разработчиками альтернативных платформ, а также снижения порога вхождения при использовании продукта на конкретной платформе. Необходимо написание библиотеки, реализующей функционал магазина на основе Playstation API.

2 Глоссарий

Термин	Определение
Токен	Специальная шифрованная строка, необходимая для взаимодействия с Xsolla
Header	Верхняя часть окна магазина; включает в себя всё до пунктов навигационного меню включительно
Footer	Нижняя часть окна магазина
Серверная интеграция	Интеграция и взаимодействие с Xsolla происходит посредством специально настроенного сервера
Упрощённая интеграция	Сервер игры отсутствует, все взаимодействия через клиента

2 Эксплуатационное назначение

Целью создания данного продукта является:

- предоставление удобного инструмента для разработчиков, который бы снизил трудозатраты по внедрению биллингового решения компании Xsolla;
- улучшение user experience конечного пользователя – инструмент должен быть нативным, требовать меньше (или исключать) дополнительные переходы на сторонние окна.

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 полный функционал

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 взаимодействиями, включающая в себя следующие экраны:

- экран виртуальных товаров;
- экран виртуальной валюты;
- экран подписок, экраны выбора платежных методов;
- экран оплаты;
- экран статуса;
- экран ошибки.

Реализовать 2 варианта взаимодействия с Playstation API (<https://secure.xsolla.com/paystation2/api>) в формате post или get запросов.

1. Серверная интеграция – передается строка токена
2. Упрощённая 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 возникло необходимости самостоятельно генерировать его.
- возможность взаимодействия с Xsolla API в сандрокс режиме. В url перед secure добавлять “sandbox-”. 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  
платёж  
ErrorCallbck // возможность обработать платёж 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.

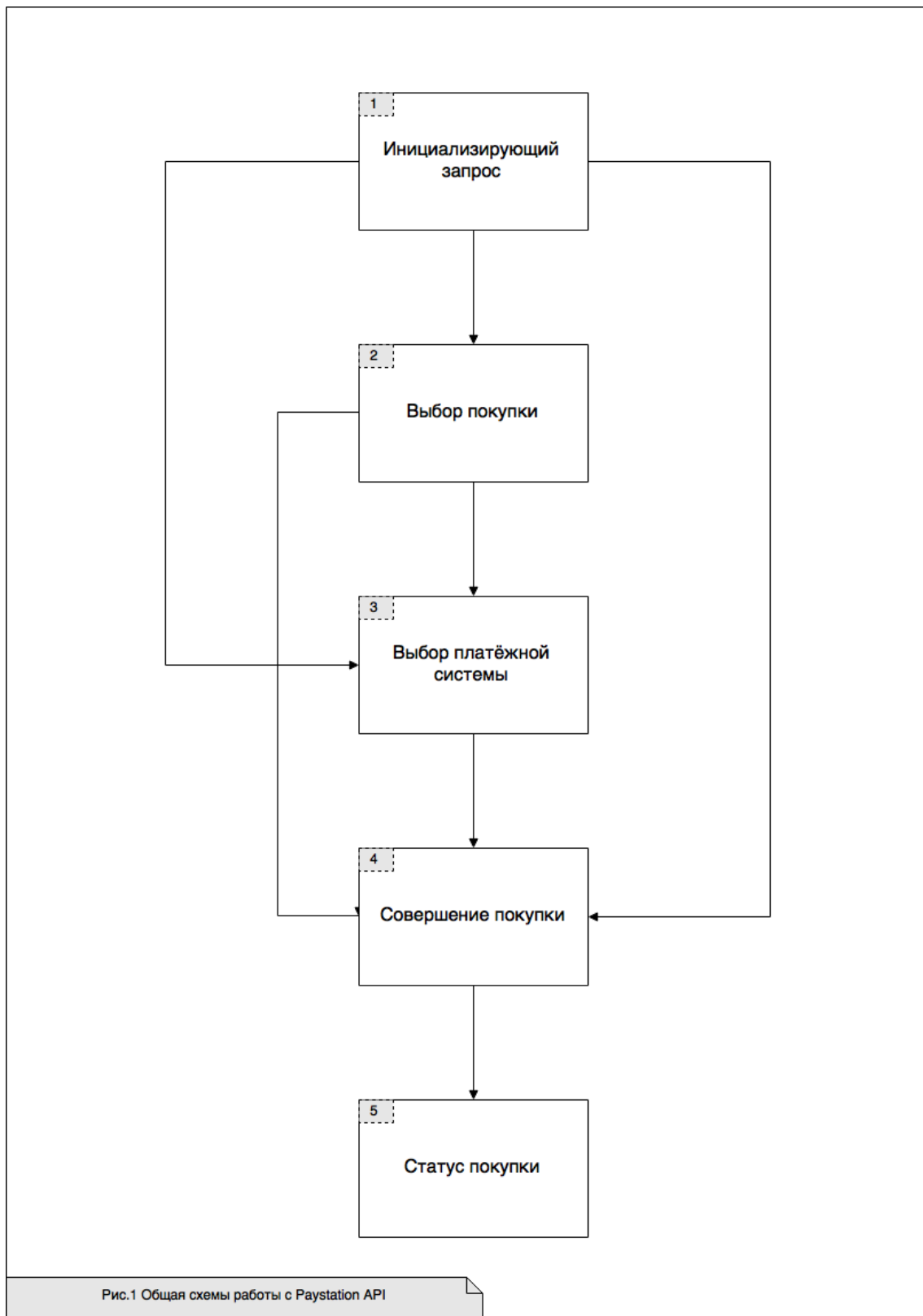
Порядок запросов и переходов описан в разделе Общая схема работы, построение интерфейса и обработка запросов описана в разделе Обработка запросов, описание дизайна находится в разделе Дизайн.

5 Общая схема работы

Общая схема взаимодействия (рис. 1) с Playstation API включает в себя 5 основных шагов:

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

Часть из этих шагов (№№2, 3) не являются обязательными и при необходимости могут быть пропущены, а именно шаги 2, 3. Вся информация, необходимая для определения схемы, может быть получена из первого шага или инициализирующего запроса. Далее рассматривается каждый из этих шагов в отдельности. Описание будет состоять из текстового описания и диаграмм, наглядно иллюстрирующих процесс работы сервиса.



5.1 Инициализирующий запрос

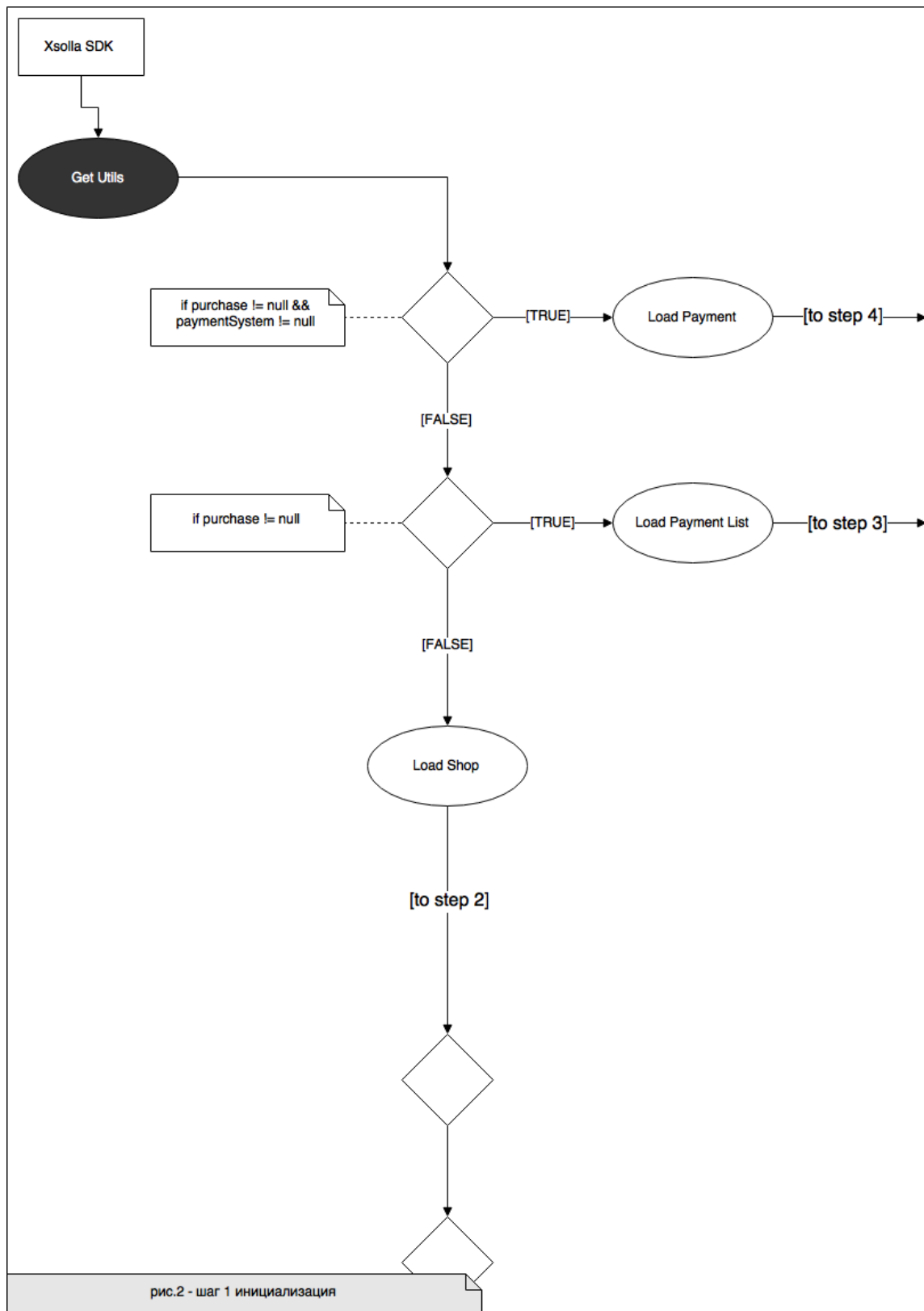
Инициализирующим запросом является запрос Utils, выходными данными которого является необходимая информация для работы разрабатываемого сервиса. Ответ на запрос

содержит в себе Информацию о пользователе, Информацию о проекте, Информацию о покупке, Настройки и Переводы.

Если после выполнения запроса получен корректный ответ от сервера, включающий в себя Информацию о покупке, становится возможным определить, была ли передана покупка("purchase"). Если покупка была передана, то необходимо проверить наличие товара ("virtual_currency", "virtual_items", "subscription" или "subscription") и платёжной системы ("payment_system").

Данные проверки позволяют определить дальнейшие шаги. Если в ответе содержится товар, то следует пропустить шаг номер 2 (рис. 1). Если содержится платёжная система, то следует пропустить шаг номер 3 (рис. 1). Схема запроса представлена на рис. 2.

В инициализирующем запросе происходит построение интерфейса системы, так как на данном этапе получены все необходимые данные для построения header и footer окна магазина.

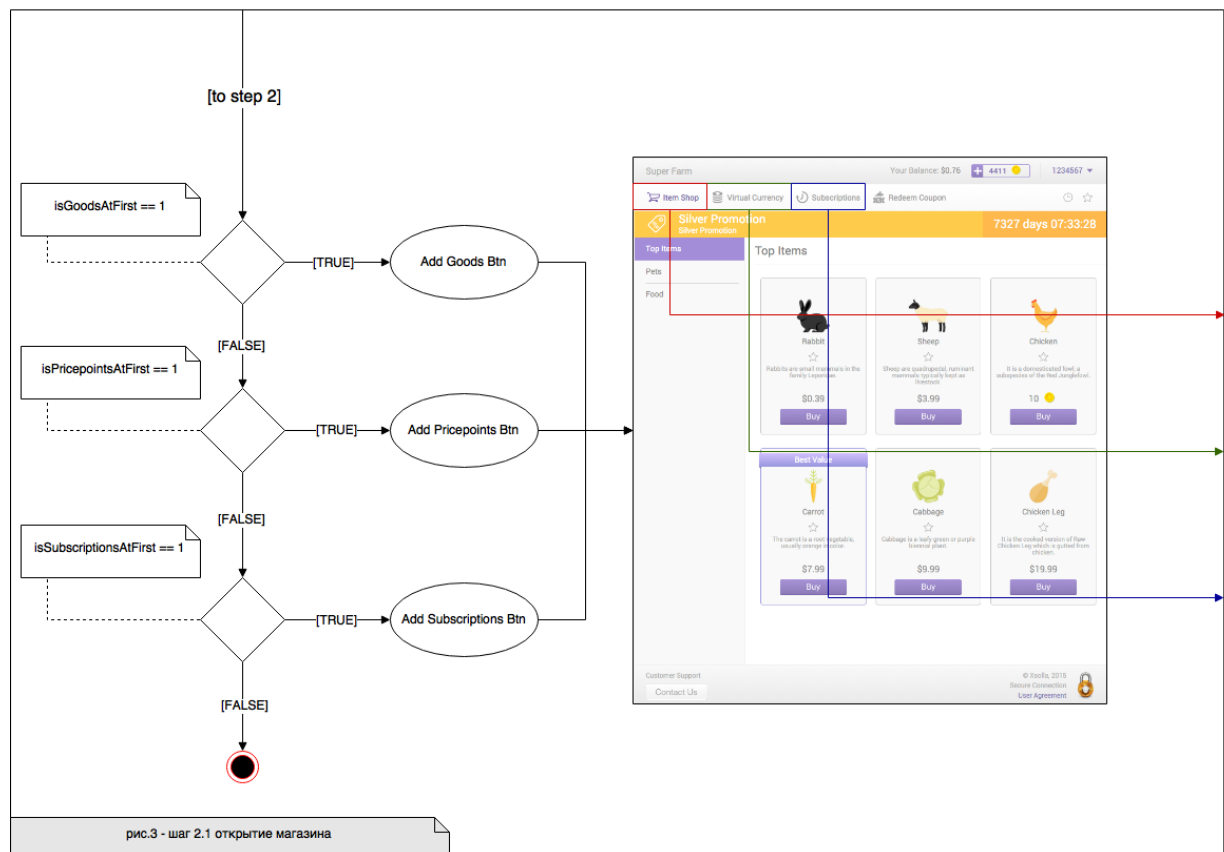


5.2 Выбор покупки

Выбор покупки — это второй шаг, в случае если перед этим не был получен товар("virtual_currency", "virtual_items", "subscription" или "subscription"). В данном запросе

предстоит построить магазин, в котором пользователь сможет выбрать понравившийся ему товар и продолжить свою покупку.

В первую очередь используется информация, полученная при помощи инициализирующего запроса. Ключевыми полями являются поля из объекта настроек("settings"): "goods_at_first", "pricepoints_at_first", "subscriptions_at_first", принимающие значения 0 или 1. В зависимости от того, что было получено на предыдущем шаге, будут отображены те или иные пункты меню Товары, Вирт валюта и Подписки соответственно (рис. 3). Та же логика соблюдается при переключении пунктов меню. В зависимости от действий пользователя при выборе какого-либо из товаров реализуется покупка за реальную валюту или покупка за виртуальную валюту.



5.2.1 Покупка за реальную валюту

В случае покупки за реальную валюту происходит переход к проверке пользовательского баланса. Если баланса достаточно, оплата продолжается через Xsolla Balance; в ином случае открывается список платёжных систем.

5.2.2 Покупка за виртуальную валюту

При покупке за виртуальную валюту реализуется другая логика (рис. 4). На первом шаге выполняется запрос Summary, ответ которого включает в себя содержимое покупки и метку, которая говорит о том, возможно ли пропустить подтверждение от пользователя. Если подтверждение может быть пропущено, то происходит переход к следующему запросу Proceed, который либо возвращает ошибку, либо не возвращает ничего. В последнем случае, платёж может быть продолжен через Xsolla Balance.

Если произошёл возврат ошибки или необходимо подтверждение от пользователя, то происходит переход на экран подтверждения платежа, на котором пользователю отображается его покупка, чекбокс “спрашивать ли подтверждение в следующий раз”. Если возврат произошёл после запроса Proceed, выдаётся сообщение об ошибке.

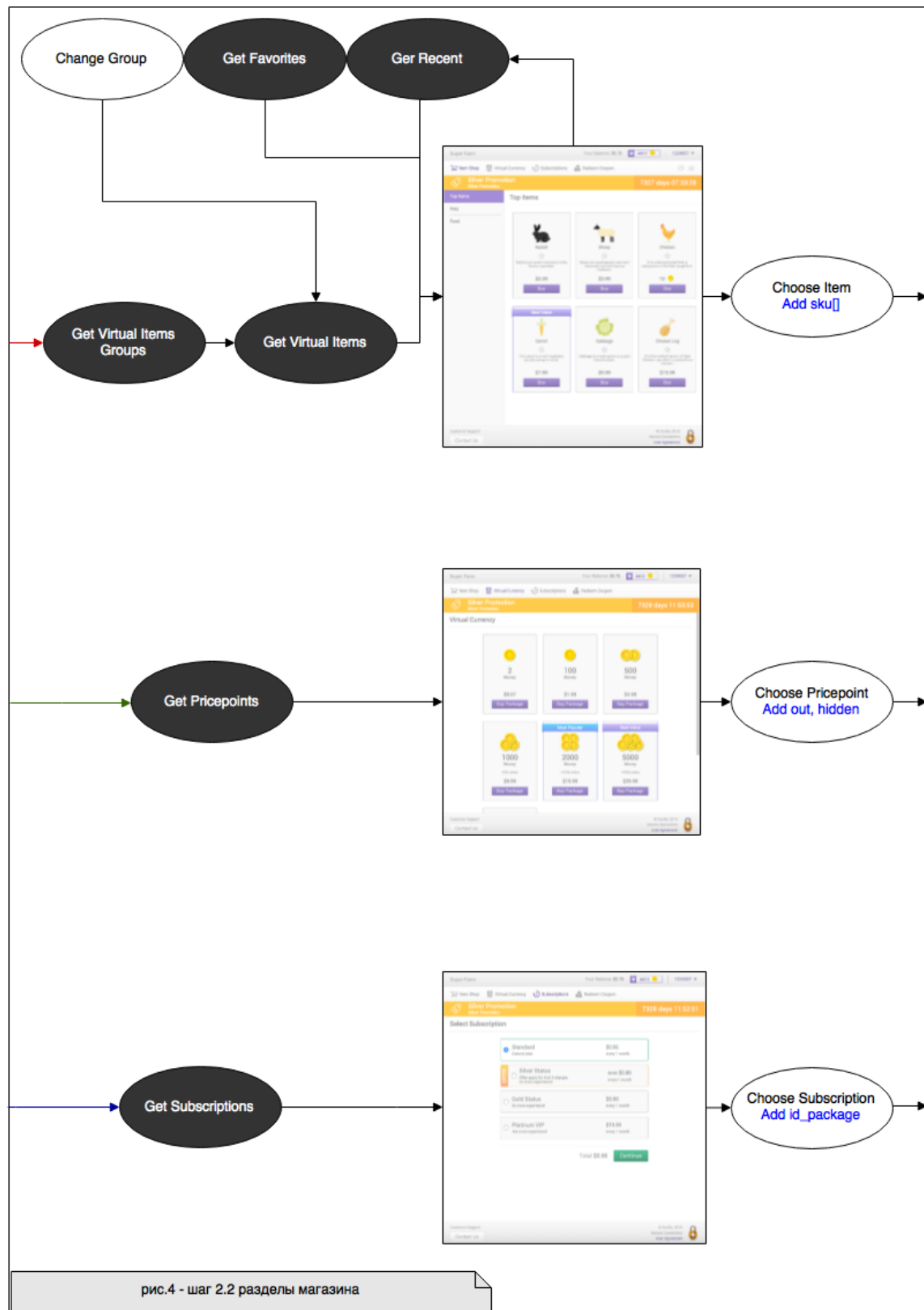


рис.4 - шаг 2.2 разделы магазина

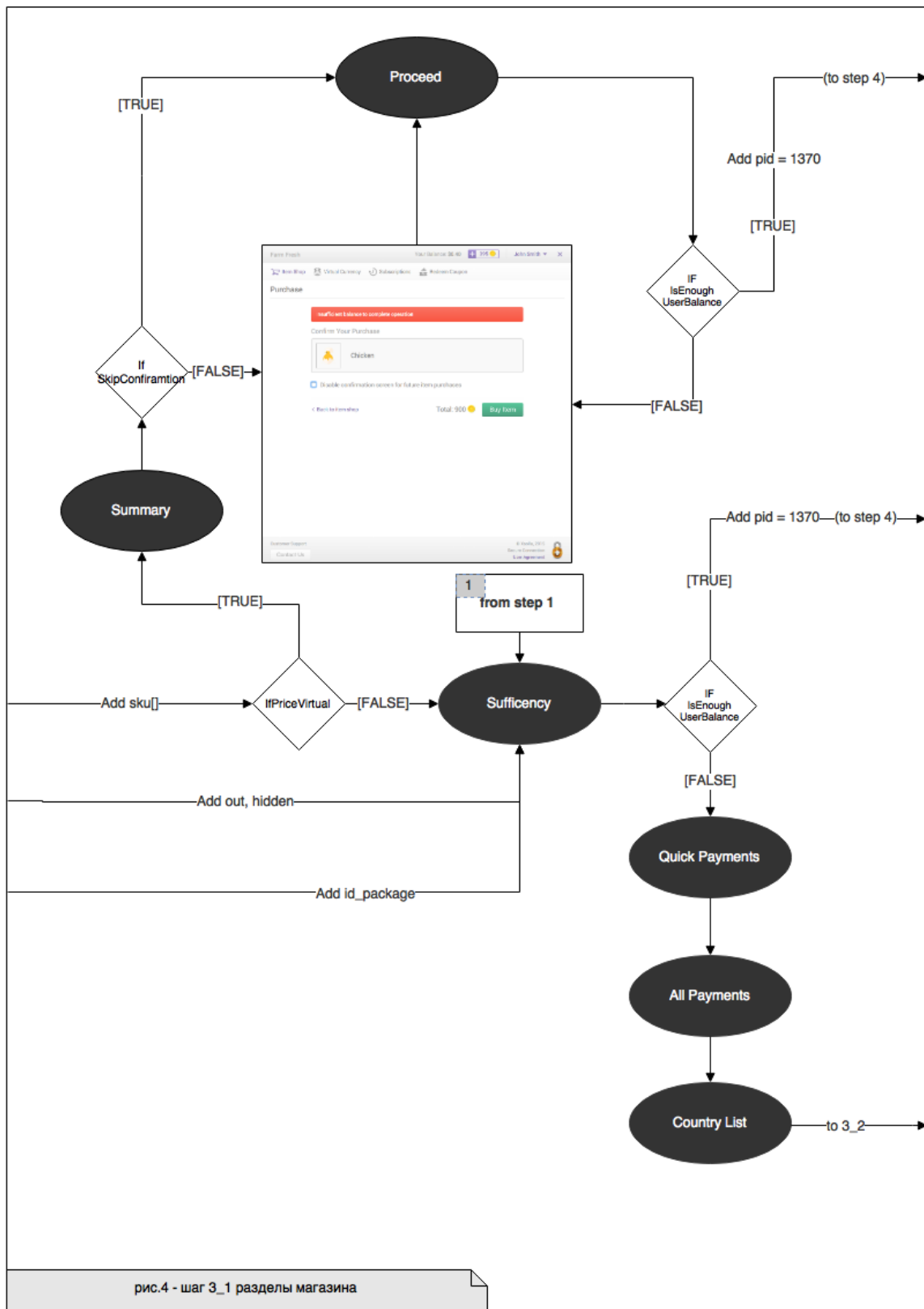


рис.4 - шаг 3_1 разделы магазина

User

