



## Разработка интерфейса терминала

### Требования и рекомендации

---

#### АННОТАЦИЯ

Описывает требования и рекомендации к процессу создания кастомизированного терминального ПО версии 7

Версия руководства: 2.2

*Руководство актуально для ТПО версий 7.6хх*

2008–2019 ООО «Софт-Лоджик», г. Барнаул, Россия

Данный документ входит в комплект поставки программных продуктов.

Права использования данного документа предусмотрены соответствующим лицензионным договором.

ООО «Софт-Лоджик»

656006, г. Барнаул, Малахова ул., дом 146в

Тел: (3852) 72-27-27

---

© *Soft-logic*

Web: <http://www.pay-logic.ru/>

Mail: [info@soft-logic.ru](mailto:info@soft-logic.ru)

---

## ОГЛАВЛЕНИЕ

<b>ИСТОРИЯ ИЗМЕНЕНИЙ.....</b>	<b>5</b>
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 0.0.0.....	5
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 0.0.1.....	5
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 0.0.2.....	6
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 2.0.....	6
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 2.1.....	7
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 2.2.....	7
<b>1 ОСНОВНЫЕ ТЕРМИНЫ.....</b>	<b>8</b>
<b>2 ПРЕДПОЛАГАЕМАЯ АУДИТОРИЯ.....</b>	<b>9</b>
<b>3 ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ JAVA FX.....</b>	<b>10</b>
3.1 ОБЩИЕ СВЕДЕНИЯ.....	10
3.2 КОМПОНЕНТЫ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА.....	11
3.3 FXML И SCENE BUILDER.....	15
3.4 СТИЛИ.....	17
3.5 ЭФФЕКТЫ.....	18
3.6 БРАУЗЕР.....	20
<b>4 ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ ИНТЕРФЕЙСА.....</b>	<b>21</b>
4.1 ТИПЫ ЭКРАНОВ И ИХ СТРУКТУРА.....	21
4.1.1 ОБЩИЕ СВЕДЕНИЯ.....	21
4.1.2 ЭКРАНЫ МЕНЮ.....	23
4.1.3 ЭКРАНЫ ВВОДА ДАННЫХ.....	25
4.1.4 ЭКРАНЫ ОПЛАТЫ.....	35
4.1.5 СПЕЦИАЛЬНЫЕ ЭКРАНЫ.....	40
4.2 ТИПЫ ДИАЛОГОВ И ИХ СТРУКТУРА.....	45
4.3 КНОПКИ.....	48
4.4 ЭЛЕМЕНТЫ ВВОДА.....	49
4.4.1 ОБЩИЕ СВЕДЕНИЯ.....	49
4.4.2 КЛАВИАТУРЫ.....	50

---

4.4.3	ТЕКСТОВЫЕ И ЧИСЛОВЫЕ ПОЛЯ.....	52
4.4.4	АВТОКОМПЛИТ.....	52
4.5	ЭЛЕМЕНТЫ НАВИГАЦИИ.....	54
4.6	РЕКЛАМА. РЕКЛАМНЫЕ МЕСТА И БАННЕРЫ.....	56
4.7	ТЕКСТОВКИ И ЛОКАЛИЗАЦИЯ.....	57
5	ТРЕБОВАНИЯ ДЛЯ РАЗРАБОТКИ ИНТЕРФЕЙСА.....	59
5.1	ЭТАПЫ СОЗДАНИЯ КАСТОМИЗИРОВАННОГО ИНТЕРФЕЙСА.....	59
5.2	СОГЛАСОВАНИЕ ПЕРЕЧНЯ ЭКРАНОВ.....	60
5.3	ПОДГОТОВКА РАБОЧИХ МАКЕТОВ И РЕСУРСОВ.....	65

**ИСТОРИЯ ИЗМЕНЕНИЙ****ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 0.0.0**

Дата публикации: 25.11.2014.

Изменение	Раздел
<b>Общие улучшения в документе:</b>	
Документ создан	-

**ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 0.0.1**

Дата публикации: 25.11.2014.

Изменение	Раздел
<b>Общие улучшения в документе:</b>	
Оформлены примеры экранов Оформлены требования к передаче макетов	-

**ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 0.0.2**

Дата публикации: 08.02.2015.

Изменение	Раздел
<b>Общие улучшения в документе:</b>	
Оформлены функциональные возможности JavaFX	-

**ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 2.0**

Дата публикации: 26.04.2015.

Изменение	Раздел
<b>Общие улучшения в документе:</b>	
Оформлены экраны, оформлены требования к составу и назначению экранов	-

**ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 2.1**

Дата публикации: 14.04.2016.

Изменение	Раздел
<b>Общие улучшения в документе:</b>	
Переоформление документа	-

**ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 2.2**

Дата публикации: 30.03.2018.

Изменение	Раздел
<b>Общие улучшения в документе:</b>	
Актуализация	-

---

## 1 ОСНОВНЫЕ ТЕРМИНЫ

**Терминал** — киоск самообслуживания, аппаратное устройство, состоящее из встроенного компьютера и оборудованное необходимым периферийным оборудованием: купюроприемником, термопринтером, сенсорным экраном и т. д.

**Терминальное ПО** — программное обеспечение, устанавливаемое на терминал, взаимодействующее с его аппаратной частью и сервером процессинга.

**Экран** — реализованный внутри терминального ПО модуль, выполняющий функции визуализации и обработки бизнес-логики определенного шага работы терминального ПО.

**Элемент** — составная часть экрана: кнопка, поле, область и т.д.

---

## 2 ПРЕДПОЛАГАЕМАЯ АУДИТОРИЯ

Документ предназначен для представителей заказчика, занимающихся непосредственно подготовкой и проектированием внешнего вида и бизнес-логики работы терминального ПО, и дизайнеров.

Документ содержит три основных раздела.

В первом — кратко рассмотрены возможности платформы, что позволяет сразу понять ограничения при создании дизайна.

Второй раздел посвящен функциональным экранам, их назначению и логике работы и позволяет понять какие экраны следует отрисовать дизайнеру.

Третий раздел посвящен требованиям, предъявляемым к дизайну и макету.

---

## 3 ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ JAVA FX

### 3.1 ОБЩИЕ СВЕДЕНИЯ

Платформа JavaFX является развитием клиентской платформы Java и позволяет разработчикам приложений легко создавать и развертывать «насыщенные» интернет-приложения. Построенная на технологии Java, платформа JavaFX предоставляет богатый набор графических и медиа-API к высокопроизводительным графическим и мультимедиа-«движкам» с аппаратным ускорением, которые упрощают разработку управляемых данными клиентских приложений. Платформа JavaFX 2 включает полный набор общедоступных API. Эти интерфейсы обеспечивают беспрецедентную свободу и гибкость для построения насыщенных клиентских приложений. Платформа JavaFX объединяет лучшие возможности платформы Java и всеобъемлющую многонаправленную медиафункциональность в интуитивно понятной и универсальной среде разработки.

---

## 3.2 КОМПОНЕНТЫ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА

Визуальные элементы управления JavaFX, доступные через JavaFX API, строятся с использованием узлов в графе сцены. Они могут в полной мере обладать визуально «насыщенными» возможностями платформы JavaFX и быть переносимы между различными платформами. Классы для построения элементов управления пользовательского интерфейса находятся в пакете `javafx.scene.control`.

Список элементов управления включает в себя типичные компоненты пользовательского интерфейса, которые использовались в более ранних версиях платформы Swing. JavaFX 2.0 SDK вводит новые элементы управления, например, `TitledPane` и `TableView`.

Классы элементов управления обеспечивают дополнительные переменные и методы, помимо существующих в классе `Control`, для поддержки типичных взаимодействий пользователей в интуитивно понятной форме. Вы можете назначить определенный стиль вашему компоненту пользовательского интерфейса с помощью каскадных листов стилей. Для некоторых необычных задач, возможно, потребуется расширить класс `Control` для создания пользовательских визуальных компонентов или использовать интерфейс `Skin`, чтобы определить новый скин для существующего элемента управления. Поскольку элементы управления из пакета `javafx.scene.control` являются расширениями класса `Node`, то они могут быть интегрированы с анимацией, преобразованиями и анимированными переходами графа сцены.



Рисунок 3.2.1 — Палитра ui компонентов FX

Рассмотрим задачу создания кнопки, применяя к ней эффект «отражение» и анимируя ее путем изменения прозрачности с максимального до минимального значения. С помощью JavaFX API можно реализовать эту задачу с помощью всего нескольких строк кода. Вы можете настроить внешний вид элементов управления путем определения собственных каскадных листов стилей. Использование CSS в приложениях JavaFX является почти таким же, как использование CSS в HTML, потому что они основаны на одной и той же спецификации CSS. Визуальные состояния элементов управления определяются в файле с расширением .css.

В дополнение к типичным элементам пользовательского интерфейса, JavaFX SDK предоставляет различные диаграммы в пакете `javafx.scene.chart`. В настоящее время поддерживаются следующие типы диаграмм: гистограмма, диаграмма областей,

график, пузырьковая диаграмма, диаграмма рассеивания, круговая диаграмма (рисунок 3.2.2).



Рисунок 3.2.2 — Примеры chart в FX

---

Любая диаграмма может содержать несколько рядов данных. В отличие от других клиентских Java-инструментариев, используя JavaFX SDK, вы можете создать диаграмму в вашем приложении, добавив всего несколько строк кода. Также можно определять различные цветовые схемы и стили, применять визуальные эффекты, обрабатывать события мыши и создавать анимацию.

Контейнеры схем расположения или панели могут быть использованы для обеспечения гибкого и динамического расположения элементов управления пользовательского интерфейса в графе сцены приложения JavaFX. JavaFX Layout API включает в себя следующие контейнерные классы, которые автоматизируют общие модели схем расположения:

1. Класс `BorderPane` размещает узлы его контента в верхней, нижней, правой, левой, или центральной области.
2. Класс `HBox` организует узлы его контента горизонтально в одну строку.
3. Класс `VBox` организует узлы его контента вертикально в один столбец.
4. Класс `StackPane` размещает узлы его контента в стеке.
5. Класс `GridPane` позволяет разработчику создать гибкую сетку из строк и столбцов, в которых размещаются узлы контента.
6. Класс `FlowPane` организует узлы его контента в горизонтальный или вертикальный «поток», обтекая указанные границы по ширине или высоте.
7. Класс `TilePane` размещает узлы его контента в ячейках одинакового размера.
8. Класс `AnchorPane` позволяет разработчикам создавать узлы-якоря для привязки к верхней, нижней или левой стороне, или в центре макета.

Для достижения желаемой структуры расположения различные контейнеры могут быть вложены.

Более подробную информацию можно получить из официальной документации:  
[http://docs.oracle.com/javafx/2/ui\\_controls/jfxpub-ui\\_controls.htm](http://docs.oracle.com/javafx/2/ui_controls/jfxpub-ui_controls.htm)

---

### 3.3 FXML И SCENE BUILDER

JavaFX Scene Builder является ключевым средством разработки для платформы JavaFX. Он обеспечивает среду визуальной разработки, которая позволяет быстро создавать графические пользовательские интерфейсы для приложений JavaFX без необходимости написания кода. Когда вы создаете макет пользовательского интерфейса, автоматически генерируется соответствующий код на языке FXML.

JavaFX Scene Builder предоставляет простой, интуитивно понятный интерфейс, который может помочь быстро создать прототип интерактивного приложения.

Целевая аудитория JavaFX Scene Builder включает:

1. Разработчиков: они могут быстро создавать прототипы пользовательского интерфейса приложения и отдельно разрабатывать логику приложения.
2. Дизайнеров: они могут быстро создавать прототипы пользовательского интерфейса без написания какого-либо кода. Могут проектировать и просматривать макет визуального интерфейса и определять его вид с помощью таблиц стилей.

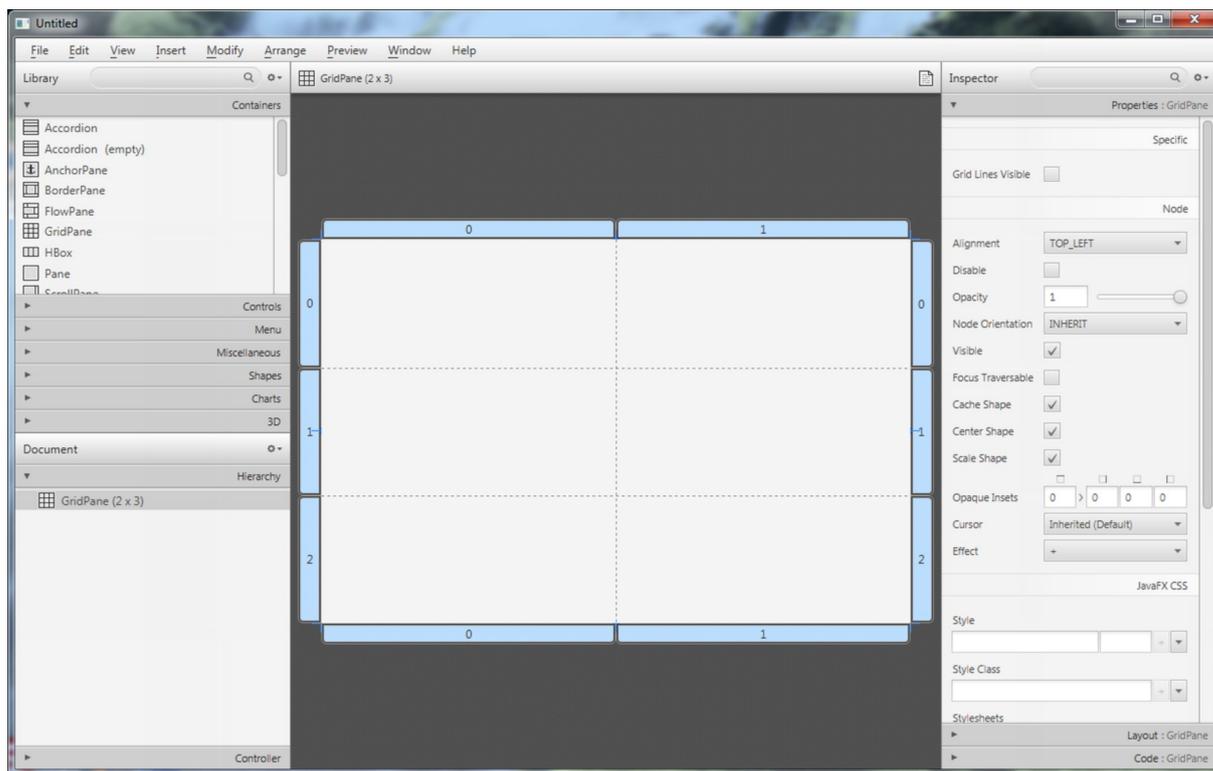


Рисунок 3.3.1 — Инструмент разработки интерфейса SceneBuilder

Более подробную информацию можно получить из официальной документации:  
[http://docs.oracle.com/javafx/scenbuilder/1/user\\_guide/jsbpub-user\\_guide.htm](http://docs.oracle.com/javafx/scenbuilder/1/user_guide/jsbpub-user_guide.htm)

## 3.4 СТИЛИ

Каскадные таблицы стилей JavaFX CSS дают возможность применять настраиваемые стили для пользовательского интерфейса приложения JavaFX без изменения исходного кода этого приложения. CSS может быть применен асинхронно к любому узлу в графе сцены JavaFX. Стили JavaFX CSS также могут быть легко назначены сцене во время выполнения, что позволяет динамически изменять внешний вид приложения.

JavaFX CSS основан на версии 2.1 спецификации W3C CSS с добавлениями от версии 3. Поддержка JavaFX CSS и расширения были разработаны, чтобы позволить корректно разбирать листы стилей JavaFX с помощью любого совместимого парсера CSS, даже не поддерживающего расширения JavaFX. Это позволяет смешивать стили CSS для JavaFX и для других целей (например, для HTML-страниц) в единой таблице стилей. Все названия свойств JavaFX начинаются с расширения поставщика «-fx-».

Более подробную информацию можно получить из официальной документации:  
<http://docs.oracle.com/javafx/2/api/javafx/scene/doc-files/cssref.html>.

## 3.5 ЭФФЕКТЫ

Каждый узел в графе сцены JavaFX с помощью классов пакета `followingjavafx.scene.transform` может быть преобразован в системе координат XY:

1. **Перемещение** — перемещение узла из одного места в другое по отношению к его первоначальной позиции.
2. **Масштабирование** — увеличение или уменьшение размера узла в зависимости от коэффициента масштабирования.
3. **Сдвиг** — поворот одной из координатных осей так, чтобы ось X и ось Y перестали быть перпендикулярными.
4. **Поворот** — поворот узла относительно указанной опорной точки сцены.
5. **Аффинное преобразование** — выполнение линейного отображения из 2-D/3-D координат в другие 2-D/3-D координаты, сохраняя при этом параллельность прямых. Этот класс используется неявно в классах, которые отвечают за другие преобразования.

Разработка «насыщенных» клиентских интерфейсов в графе сцены JavaFX включает в себя использование визуальных эффектов или эффектов, улучшающих внешний вид приложений JavaFX в реальном времени. Например:

1. **Тень** — отображает тень данного контента за контентом, к которому применяется эффект.
2. **Отражение** — отображает отраженную версию контента под ним.
3. **Освещение** — имитирует источник света, падающего на данный контент, и позволяет придать плоскому объекту более реалистичный трехмерный вид.

---

Следует учитывать, что использование эффектов сильно увеличивает нагрузку на процессор и видеокарту и тем самым повышает требования к аппаратной платформе.

---

## 3.6 БРАУЗЕР

Встроенный браузер JavaFX представляет собой новый компонент пользовательского интерфейса JavaFX, который обеспечивает веб-просмотрщик и полную браузерную функциональность посредством API. Этот компонент Web Engine основан на WebKit, который является «движком» веб-браузера с открытым исходным кодом, поддерживающим HTML5, CSS, JavaScript, DOM и SVG, а также обработку любых JavaScript-команд и обработку событий. Основная функция — это использование динамического и интерактивного веб-контента.

Более подробную информацию можно получить из официальной документации <http://docs.oracle.com/javafx/2/webview/jfxpub-webview.htm>.

## 4 ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ ИНТЕРФЕЙСА

### 4.1 ТИПЫ ЭКРАНОВ И ИХ СТРУКТУРА

#### 4.1.1 ОБЩИЕ СВЕДЕНИЯ

В этом и следующих разделах в качестве примера рассматривается стандартный интерфейс ТПО версии 7. Рассматриваются его экраны, их назначение, функциональность и внешний вид в текущем дизайне. Рассмотренные примеры описывают логику работы терминала и должны служить основой при разработке собственного дизайна ТПО.

Тип экрана напрямую зависит от своего прямого функционального назначения. Все типы экранов можно разделить на следующие группы:

1. Экраны меню.
2. Платежные экраны.
3. Специальные экраны.

Для стандартного интерфейса blues принята следующая структура экрана (рисунок 4.1.1.1):

1. Сквозной баннер.
2. Заголовок.
3. Область контента.
4. Подвал.

Заголовок экрана в общем случае всегда содержит логотип платежной системы, сообщение экрана (приветственное сообщение, название сервиса и т. п.). Так же заголовок может содержать специфические элементы, характерные для конкретных типов экранов, например клавишу переключения языка, логотип сервиса и т. п.

Подвал экрана в общем случае содержит элементы навигации ( кнопки назад, выход, далее и т.п), так же может содержать кнопки вызова специальных экранов.

Сквозной баннер — предназначен для показа баннерной рекламы. «Сквозной» — означает, что он присутствует на всех без исключения экранов и его содержимое (смена содержимого) никак не связано с процессом смены экранов в ходе работы терминального ПО: смена баннера выполняется асинхронно и независимо.

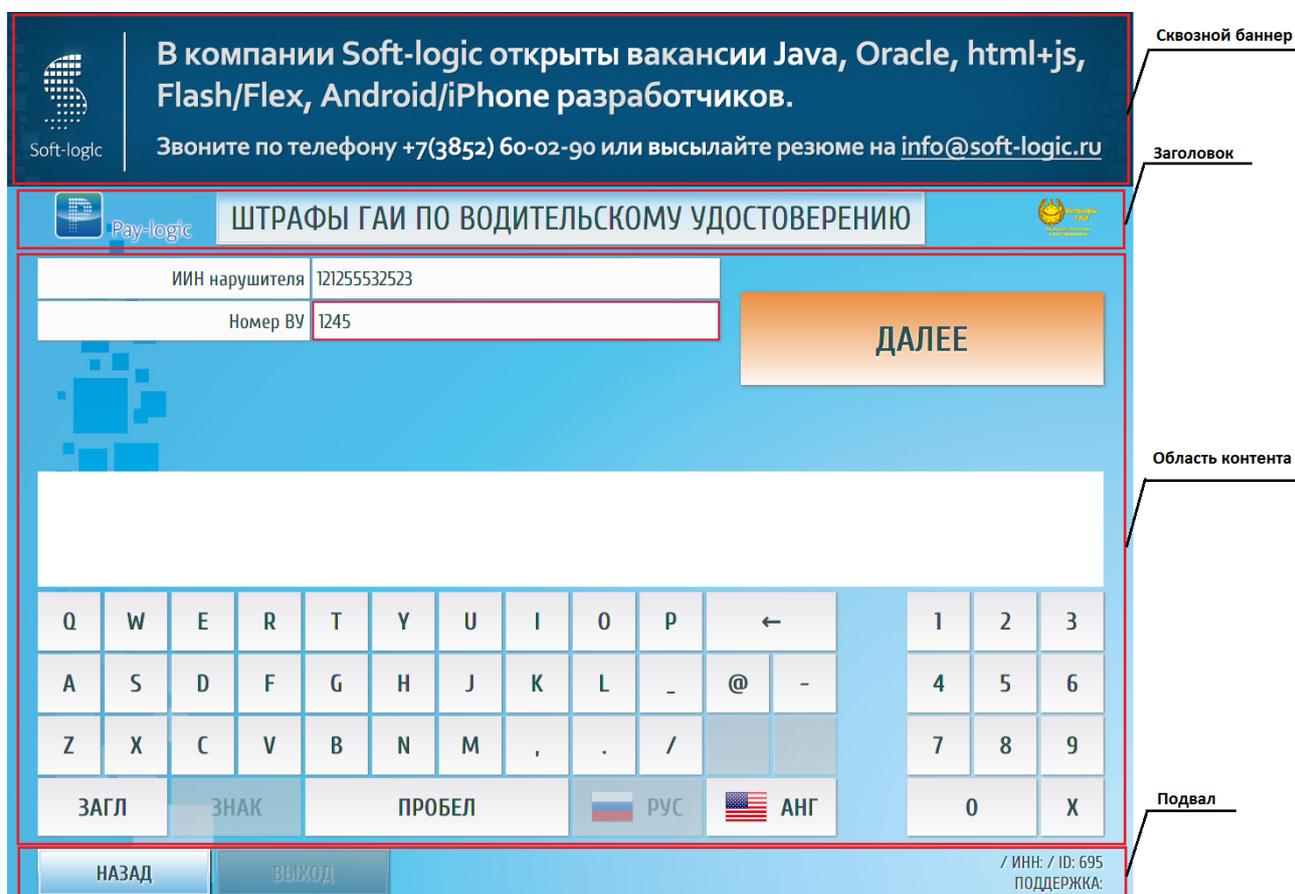


Рисунок 4.1.1.1 — Общая структура экрана

Разумеется, структура экранов может быть спроектирована дизайнером и иной. Например, может не быть сквозного баннера или быть больше/меньше структурных элементов, но структура должна быть единой для всех экранов терминального ПО.

#### 4.1.2 ЭКРАНЫ МЕНЮ

Экраны меню предоставляют функционал отображения групп сервисов, сервисов и горячих клавиш.

В стандартном интерфейсе blues существуют два типа экранов меню:

1. Экран выбора группы (тип group) — рисунок 4.1.2.1.
2. Экран выбора сервиса (тип service) — рисунок 4.1.2.2.

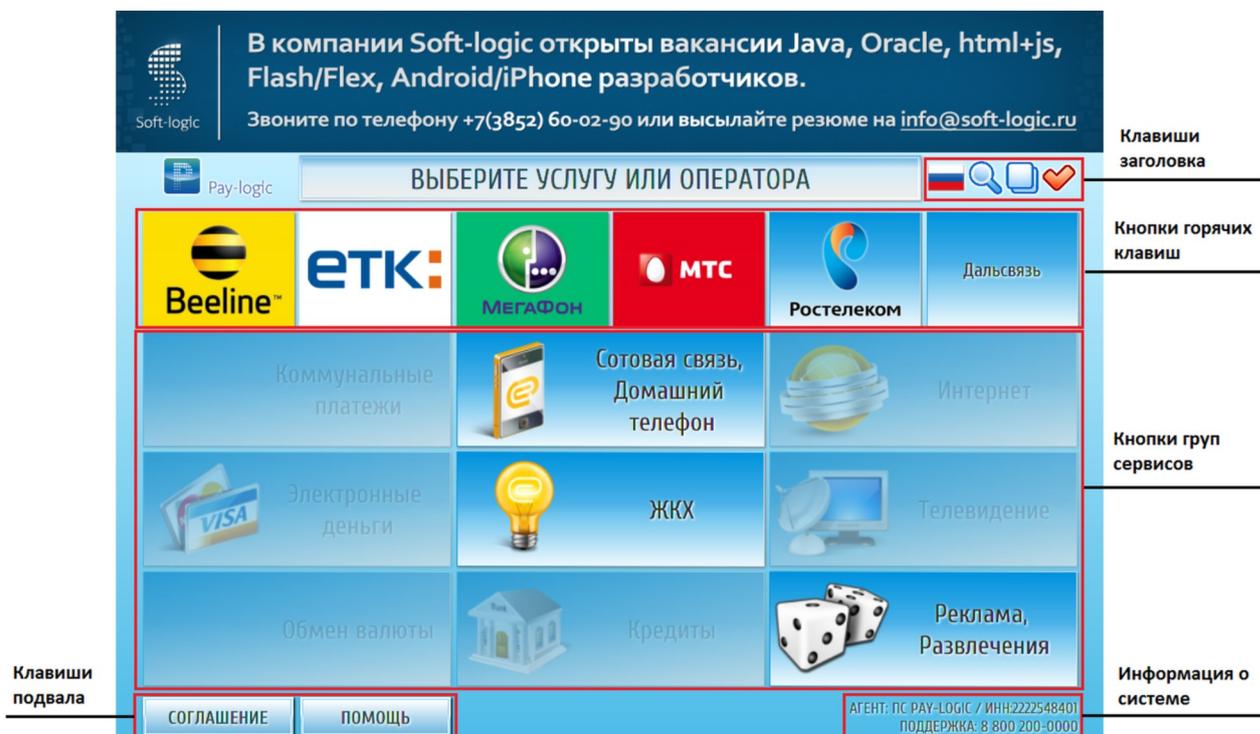


Рисунок 4.1.2.1 — Экран выбора группы

Экран выбора группы в интерфейсе blues содержит область горячих клавиш, область кнопок групп. Так же в заголовке доступны клавиша переключения языка, клавиша перехода на экран «Поиск сервиса», клавиша перехода на экран «Копия чека» и клавиша перехода на экран «Статус платежа». В подвале имеются кнопки перехода в разделы помощи и оферты, а так же информация о платежной системе и телефон технической поддержки.

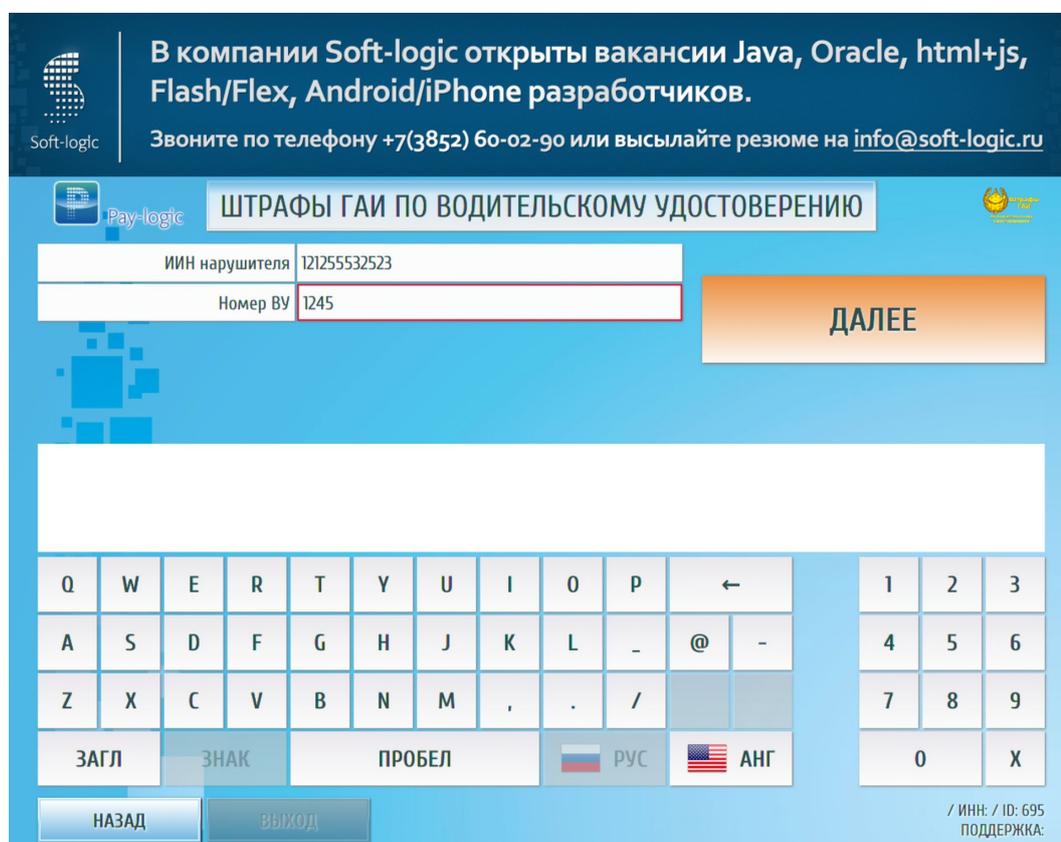


Рисунок 4.1.2.2 — Экран выбора сервиса

Экран выбора сервиса отображает кнопки выбора групп и сервисов одинаково, не группируя, согласно порядку сортировки, установленному в кабинете. В подвале имеется навигация, позволяющая вернуться в предыдущее меню, выйти в корневое меню, а также клавиши пролистывания контента.

#### 4.1.3 ЭКРАНЫ ВВОДА ДАННЫХ

Экраны ввода данных предназначены для ввода данных и отображения информации, непосредственно связанных с процессом оплаты сервиса, а именно с вводом реквизитов, прохождением платежного сценария и т. д. **Групповой экран ввода.** Служит для ввода нескольких полей и данных разных типов (тестовых, цифровых, даты, списков), рисунок 4.1.3.1.



В компании Soft-logic открыты вакансии Java, Oracle, html+js, Flash/Flex, Android/iPhone разработчиков.  
Звоните по телефону +7(3852) 60-02-90 или высылайте резюме на [info@soft-logic.ru](mailto:info@soft-logic.ru)

Soft-logic Pay-logic ШТРАФЫ ГАИ ПО ВОДИТЕЛЬСКОМУ УДОСТОВЕРЕНИЮ

ИНН нарушителя 121255532523  
Номер ВУ 1245

ДАЛЕЕ

Q W E R T Y U I O P ←  
A S D F G H J K L \_ @ -  
Z X C V B N M , . /  
ЗАГЛ ЗНАК ПРОБЕЛ РУС АНГ  
НАЗАД ВЫХОД

1 2 3  
4 5 6  
7 8 9  
0 X

/ ИНН: / ID: 695  
ПОДДЕРЖКА:

Рисунок 4.1.3.1 — Групповой экран ввода

Групповой экран содержит либо полную клавиатуру, либо всплывающую клавиатуру: в этом случае вид клавиатуры зависит от типа поля (число, текст и т.д.). Всплывающая клавиатура описана ниже.

**Экран цифрового ввода.** Служит для ввода цифрового значения (номера телефона, номера договора, суммы и т.п.). Содержит цифровую клавиатуру, рисунок 4.1.3.2.

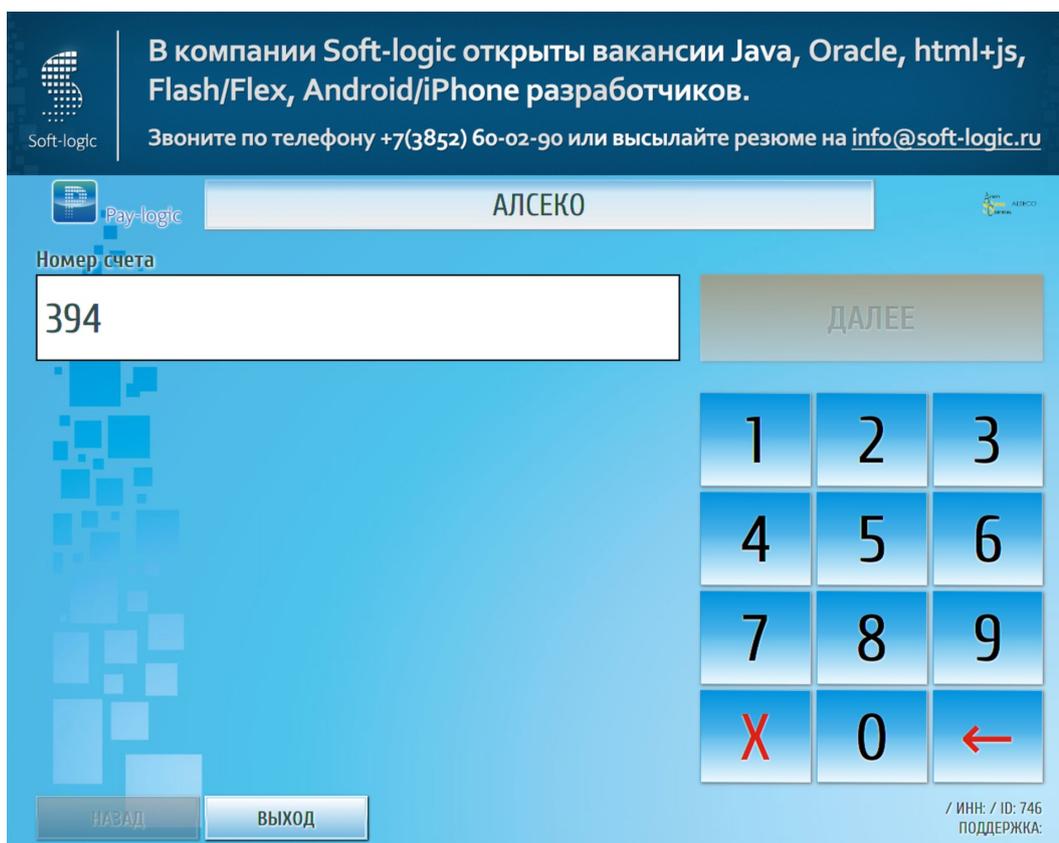


Рисунок 4.1.3.2 — Экран цифрового ввода

**Экран селектора.** Селектор — элемент ввода, предполагающий пользователю сделать выбор из нескольких predetermined значений. Например, выбрать свой тарифный план или интересующий рейс. Селекторы могут быть представлены в нескольких вариантах: текстовый селектор, селектор в виде списка значений, селектор в виде таблицы, карточный селектор, селектор номиналов.

*Текстовый селектор* приведен на рисунке 4.1.3.3. Компонент предполагает выбор одного из небольшого числа вариантов. Число вариантов обычно predetermined и не меняется.

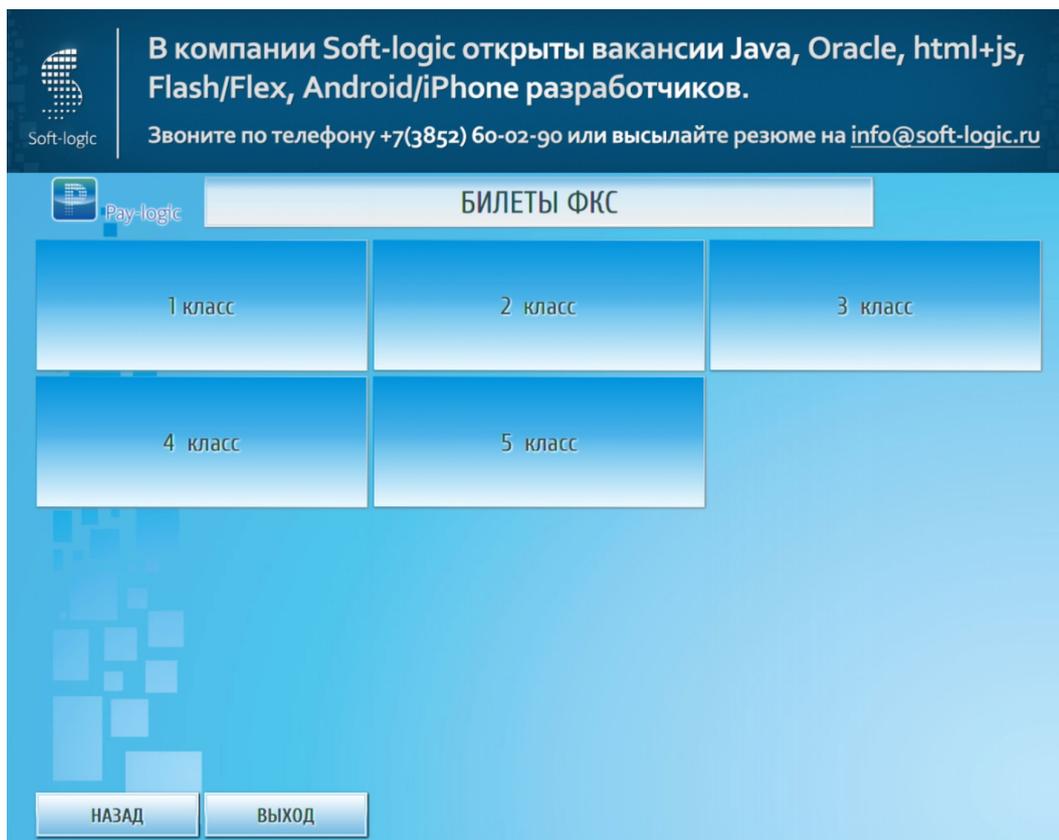


Рисунок 4.1.3.3 — Текстовый селектор

Селектор в виде списка значений применяется тогда, когда вариантов выбора либо много, либо переменное количество, а в названии варианта нужно вывести длинный текст, рисунок 4.1.3.4.

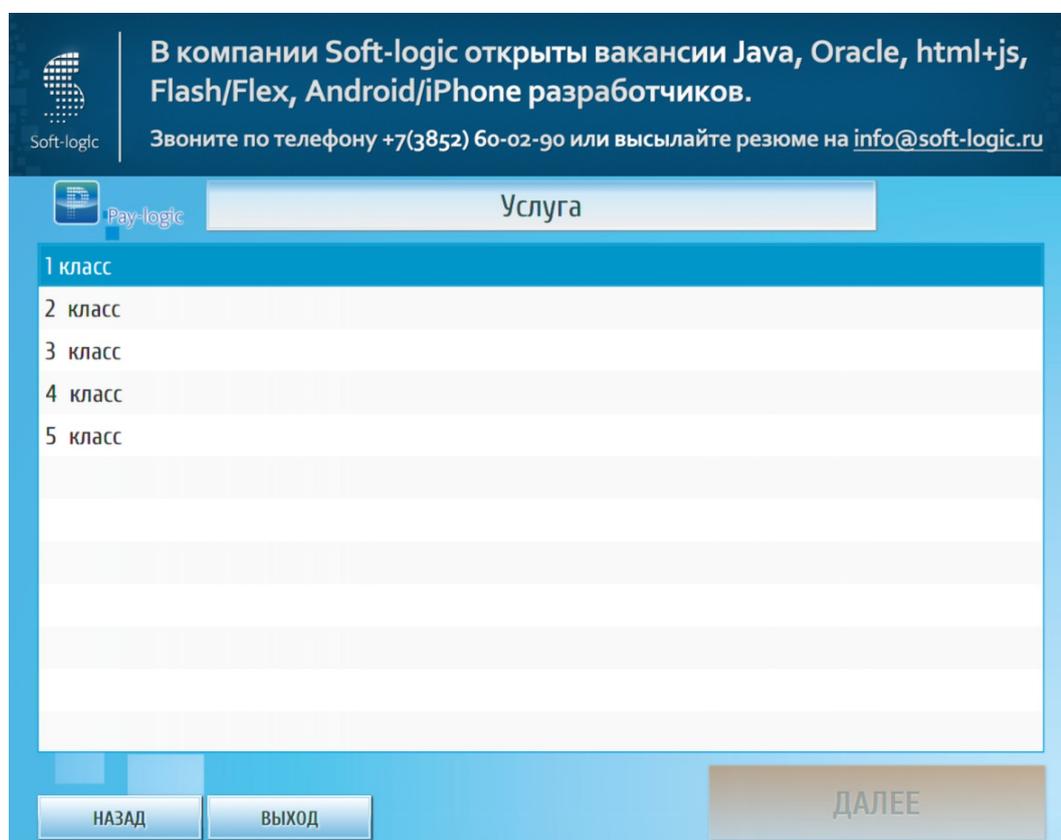
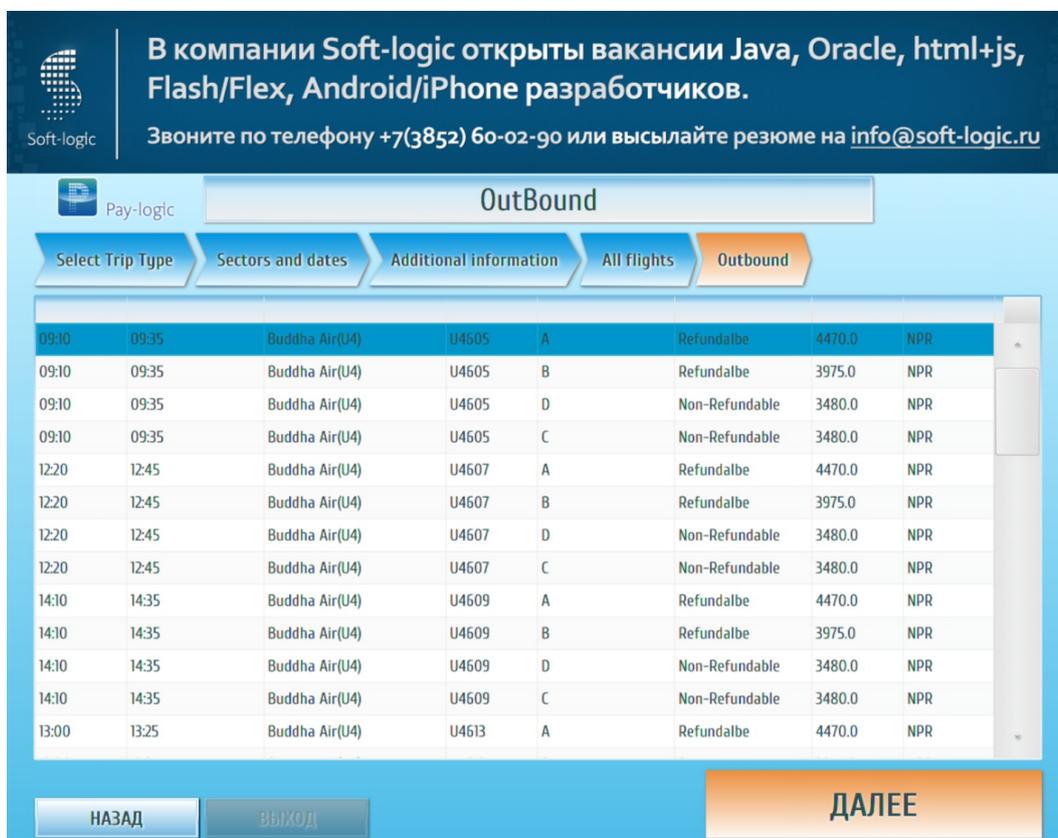


Рисунок 4.1.3.4 — Селектор в виде списка значений

Селектор в виде таблицы (рисунок 4.1.3.5) применяется в случае, если вариант выбора — строка, которая содержит в себе разнородные данные. Например, для отображения списка рейсов — пункт отправления, время отправления, цена.



09:10	09:35	Buddha Air(U4)	U4605	A	Refundalbe	4470.0	NPR
09:10	09:35	Buddha Air(U4)	U4605	B	Refundalbe	3975.0	NPR
09:10	09:35	Buddha Air(U4)	U4605	D	Non-Refundable	3480.0	NPR
09:10	09:35	Buddha Air(U4)	U4605	C	Non-Refundable	3480.0	NPR
12:20	12:45	Buddha Air(U4)	U4607	A	Refundalbe	4470.0	NPR
12:20	12:45	Buddha Air(U4)	U4607	B	Refundalbe	3975.0	NPR
12:20	12:45	Buddha Air(U4)	U4607	D	Non-Refundable	3480.0	NPR
12:20	12:45	Buddha Air(U4)	U4607	C	Non-Refundable	3480.0	NPR
14:10	14:35	Buddha Air(U4)	U4609	A	Refundalbe	4470.0	NPR
14:10	14:35	Buddha Air(U4)	U4609	B	Refundalbe	3975.0	NPR
14:10	14:35	Buddha Air(U4)	U4609	D	Non-Refundable	3480.0	NPR
14:10	14:35	Buddha Air(U4)	U4609	C	Non-Refundable	3480.0	NPR
13:00	13:25	Buddha Air(U4)	U4613	A	Refundalbe	4470.0	NPR

Рисунок 4.1.3.5 — Селектор в виде таблицы

*Селектор карт* предназначен для продажи номиналов (пинов) и представляет собой вариант графического селектора: кнопки, содержащие изображение и текст. Текст может включать стоимость приобретаемой карты (необязательно). Можно рассматривать его как графический кнопочный селектор.

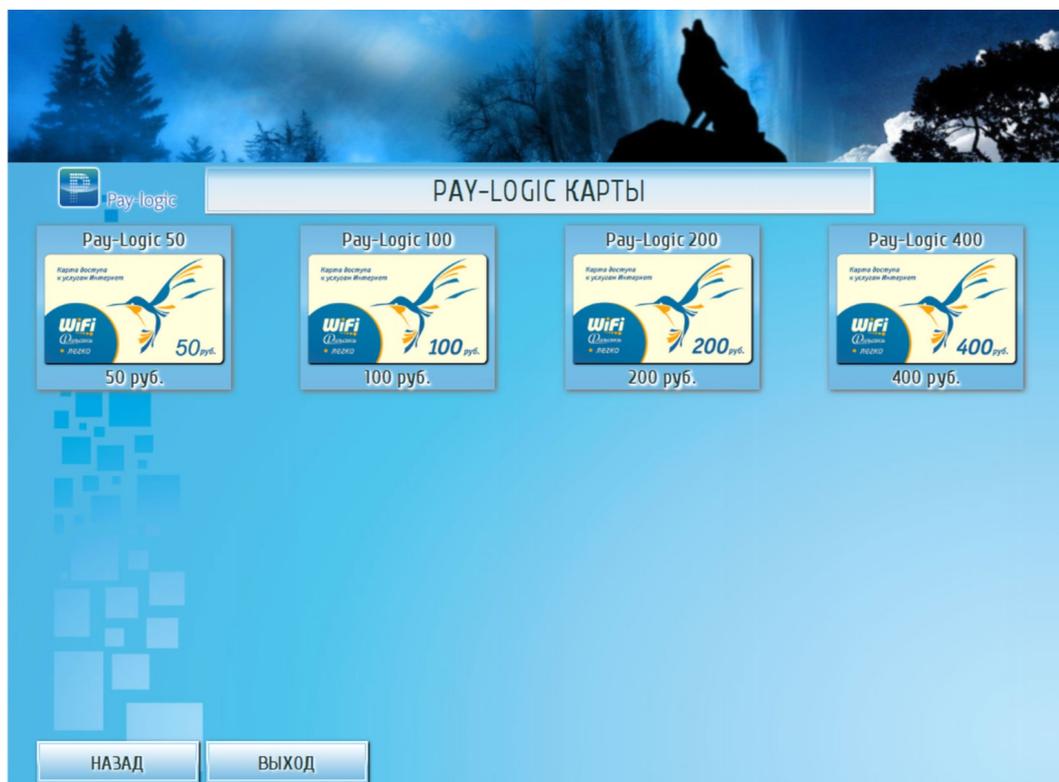


Рисунок 4.1.3.6 — Селектор карт

**Экран выбора даты** (рисунок 4.1.3.7). Представляет собой удобный календарь, и служит для выбора даты (дня) с возможностью листания года/месяца. Удобен для указания близлежащих дат, но крайне неудобен для указания далеко отстоящих дат (например, даты договора/даты окончания действия и т.д.)

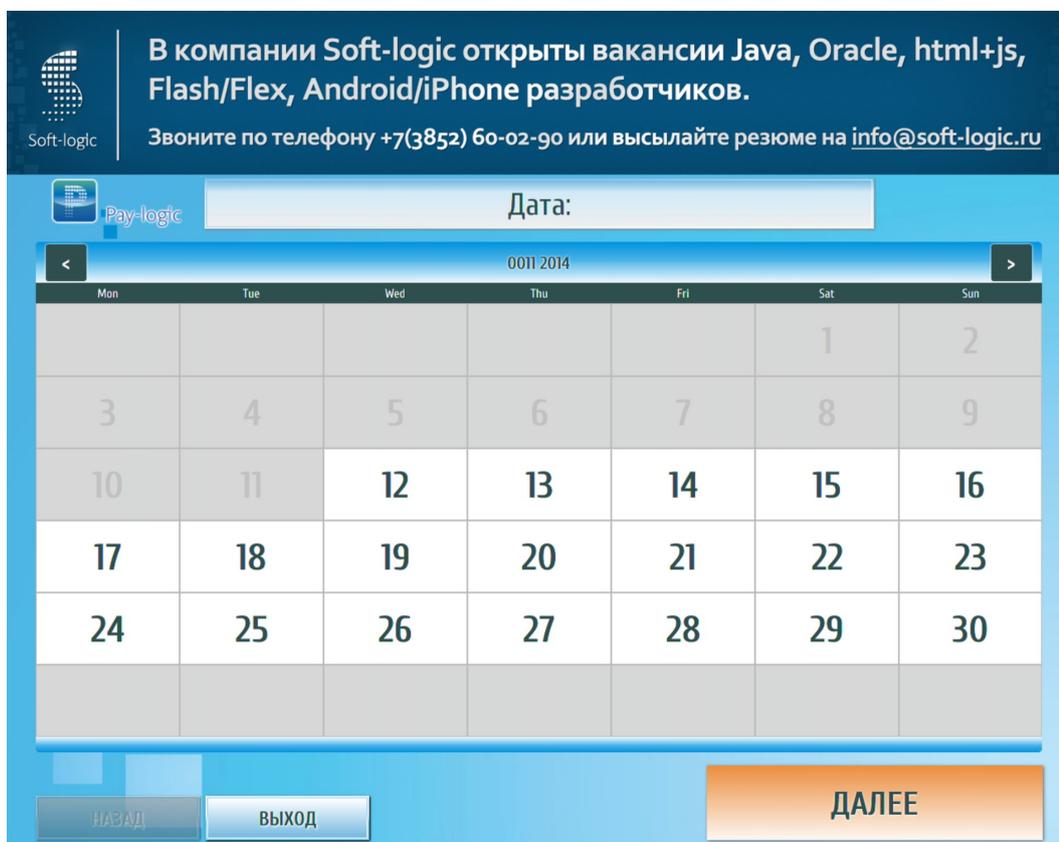


Рисунок 4.1.3.7 — Экран выбора даты

Для указания далеко отстоящих дат целесообразно применять текстовое поле, в котором дата просто вводится.

В отдельных случаях требуется указывать не дату, а год/месяц. В таком случае целесообразна разработка отдельного экрана, на котором последовательно из двух выпадающих списков будет предоставлен выбор года и затем месяца.

**Экран Корзины.** Используется при работе со сложными сервисами, когда платательщик самостоятельно составляет итоговый перечень услуг (возможностей) к оплате, в т. ч. указывая по каждому пункту желаемую сумму или иные атрибуты (рисунок 4.1.3.8).

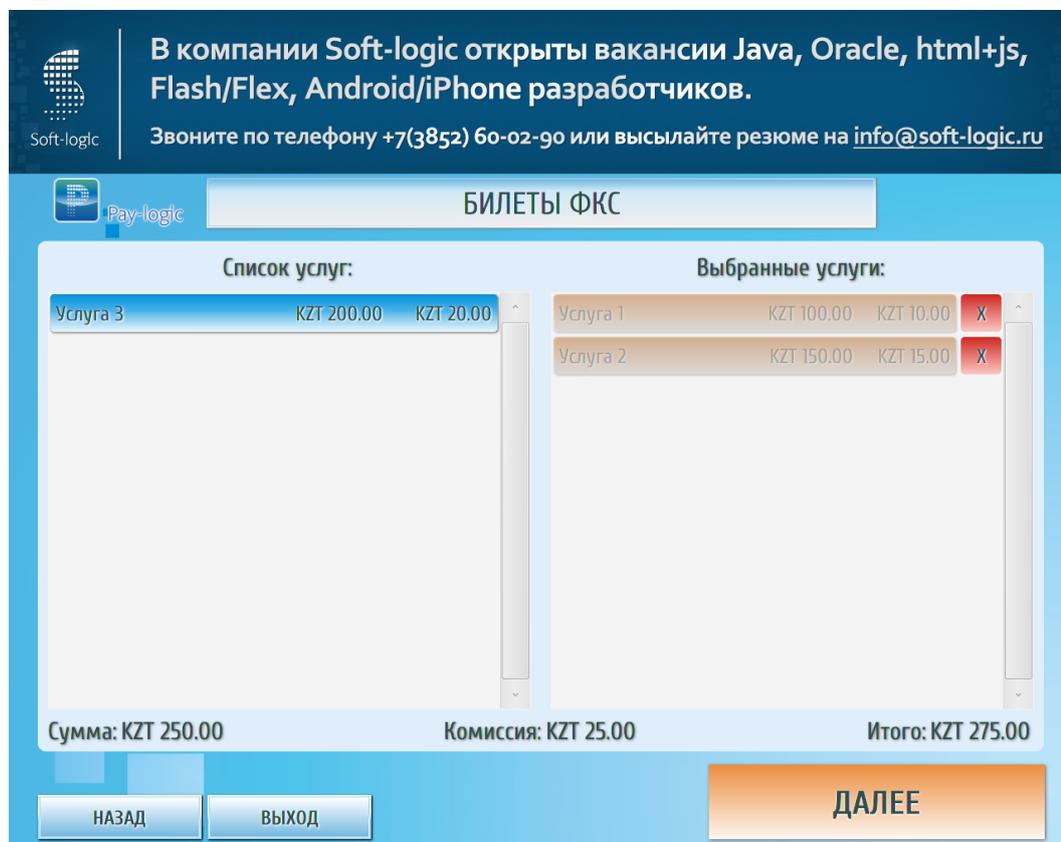


Рисунок 4.1.3.8 — Экран корзины

**Экран подтверждения.** Предназначен для отображения данных, введенных пользователем либо полученных с сервера, с целью проверки правильности данных плательщиком. Обычно такой экран показывается после ввода всех данных, но перед экраном внесения средств. Существует в двух вариантах: с одним полем и с множеством полей.

*Вариант с одним полем* применяется для оплаты простых сервисов наподобие сотовой связи, рисунок 4.1.3.9.

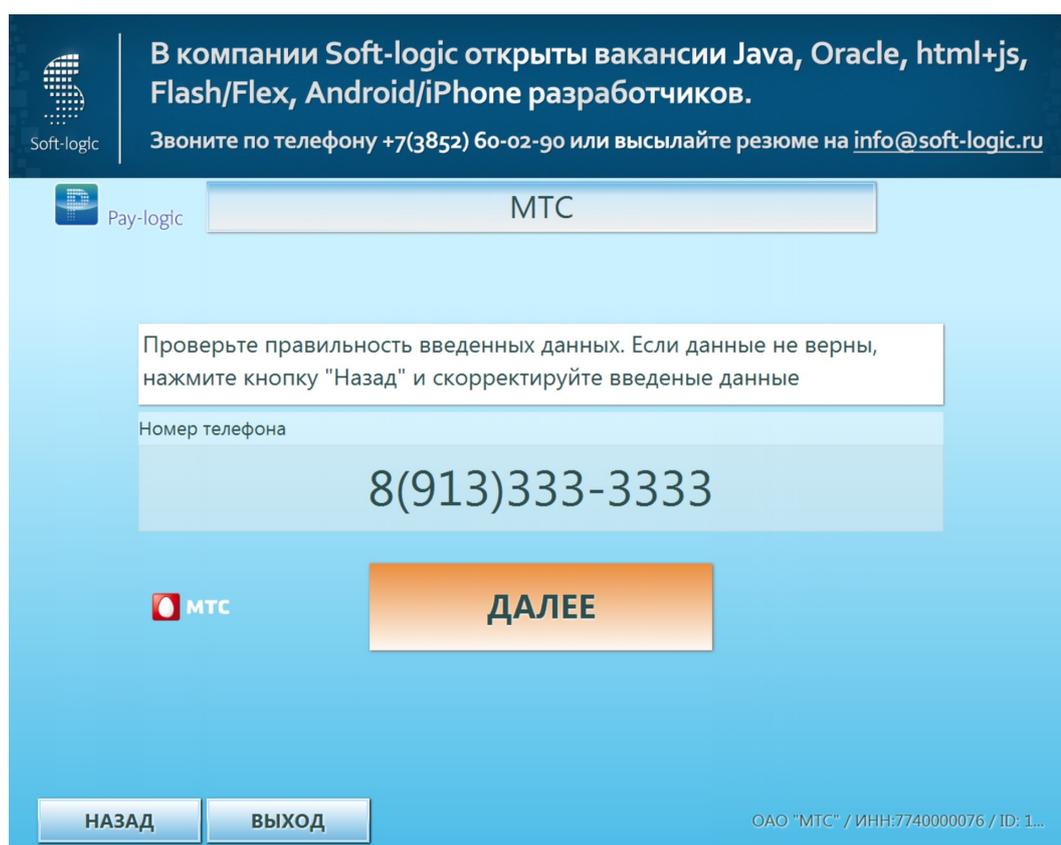


Рисунок 4.1.3.9 — Экран подтверждения

Вариант с несколькими полями применяется для всех остальных услуг, где для проверки нужно показывать несколько реквизитов (рисунок 4.1.3.10).

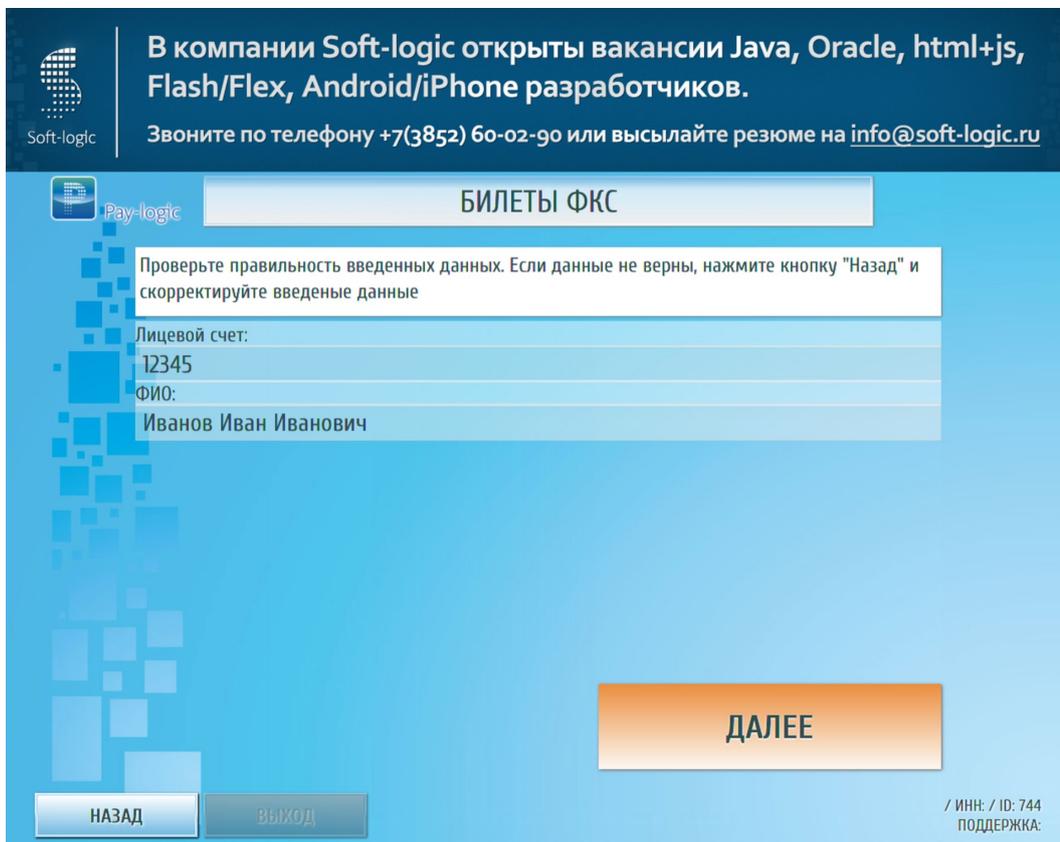


Рисунок 4.1.3.10 — Экран подтверждения с несколькими полями

#### 4.1.4 ЭКРАНЫ ОПЛАТЫ

Экраны оплаты предназначены для выполнения шагов по внесению денег, завершению оплаты, получению сдачи и иных действий, сопровождающих этот процесс.

**Экран внесения средств** (рисунок 4.1.4.1). Отображает требуемую сумму, внесенную сумму, рассчитанную комиссию и иные возможности оплаты. Основная функция — обеспечить внесение средств и информирование о суммах. Экран позволяет вернуться в сценарий (если деньги не были внесены), отменить платеж и получить деньги в виде сдачи, использовать ранее полученный ваучер сдачи и т. д.

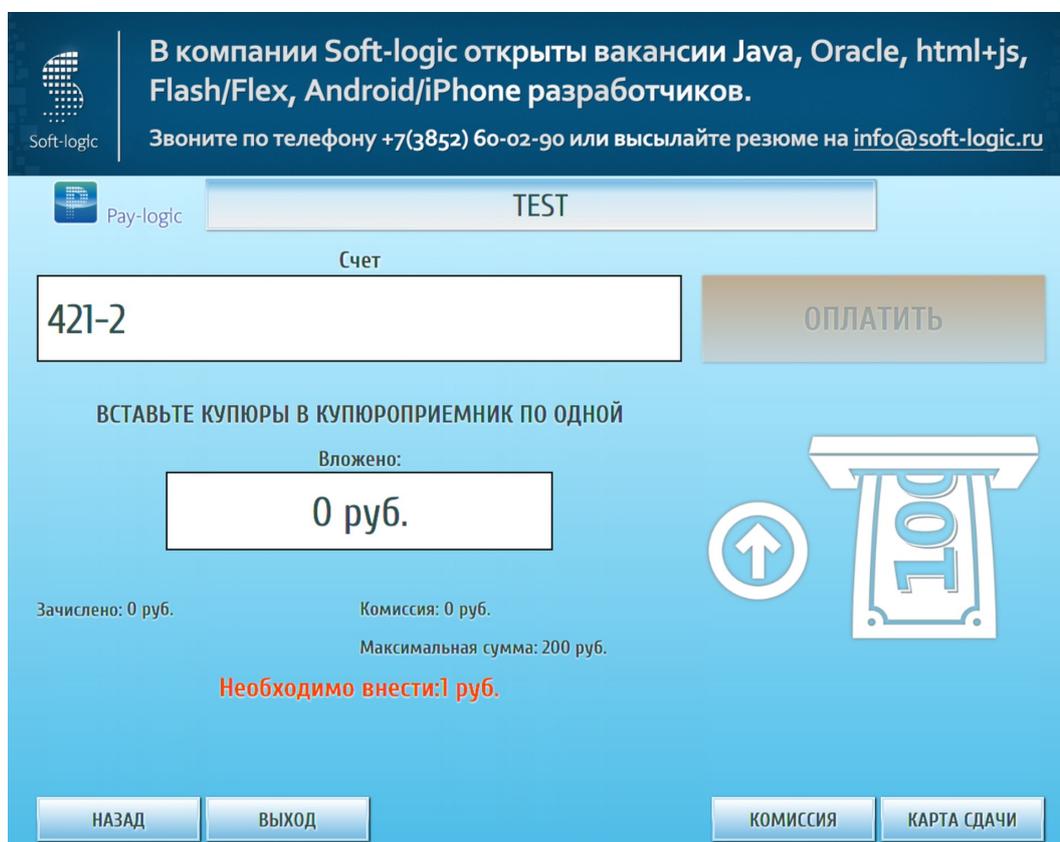


Рисунок 4.1.4.1 — Экран внесения средств

**Экран информации о комиссии.** Отображает развернутую информацию о комиссии на текущем сервисе в виде таблицы со ступенями, шагами и ставками комиссии — согласно текущих настроек процессинга. Экран в стандартном интерфейсе представляет модальное диалоговое окно (рисунок 4.1.4.2).

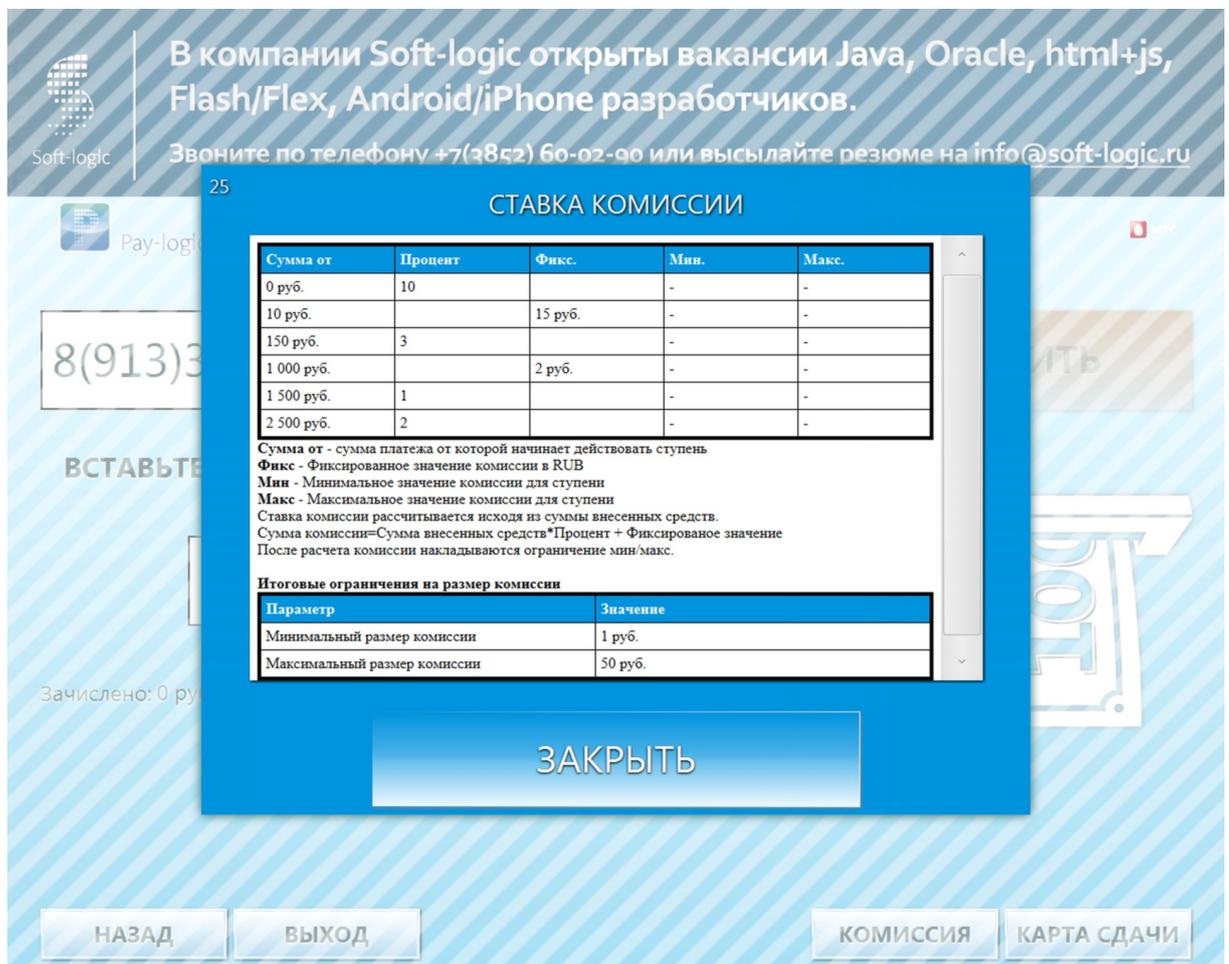


Рисунок 4.1.4.2 — Экран информации о комиссии

**Экран активации карты сдачи (ваучера сдачи).** Служит для ввода и верификации пин-кода карты сдачи, полученной плательщиком ранее. Также является диалоговым (рисунок 4.1.4.3).



Рисунок 4.1.4.3 — Экран активации карты сдачи (ваучера сдачи)

**Экран зачисления сдачи.** Представляет собой диалог выбора варианта выдачи сдачи: каким образом плательщик сможет получить сдачу, если таковая у него осталась. В стандартном интерфейсе — это диалоговое модальное окно, но в разрабатываемом интерфейсе он может быть заменен на полноценный экран (рисунок 4.1.4.4).

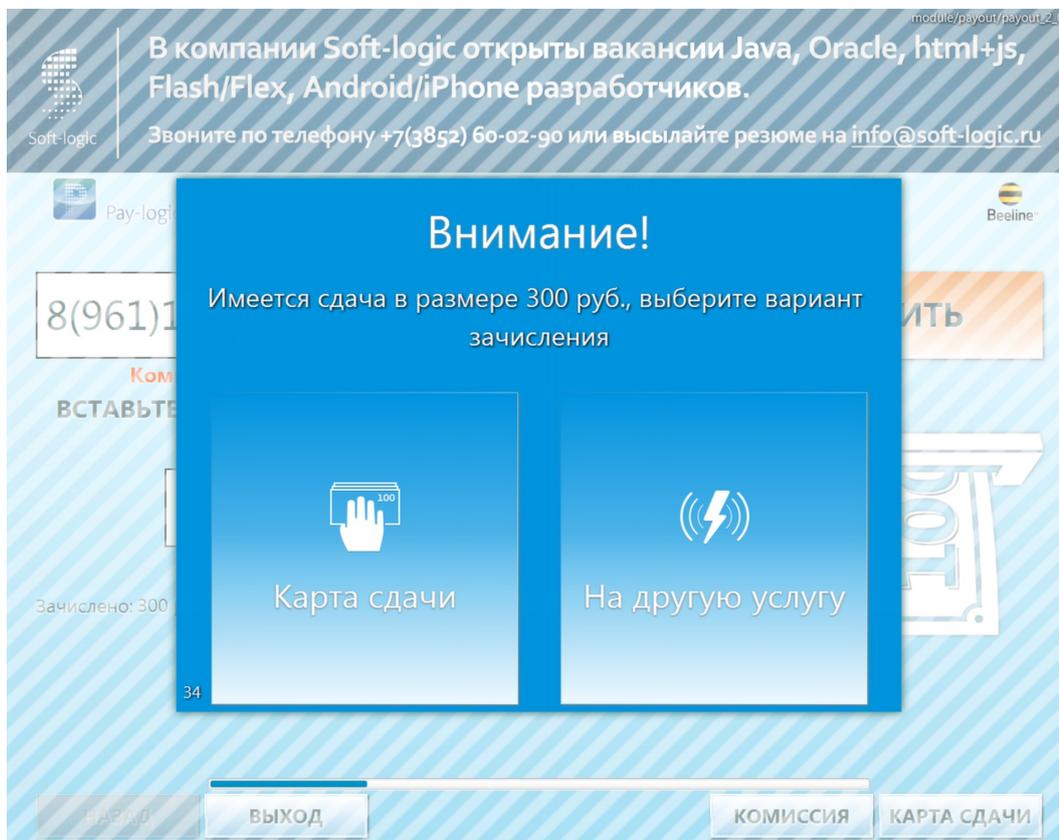


Рисунок 4.1.4.4 — Экран зачисления сдачи

**Экран завершения платежа.** В стандартном интерфейсе — диалог статуса завершения, показывающий факт успешного проведения платежа или ошибки, и благодарность за использование системы. При показе этого диалога обычно печатается чек. Также может быть заменен экраном (рисунок 4.1.4.5).

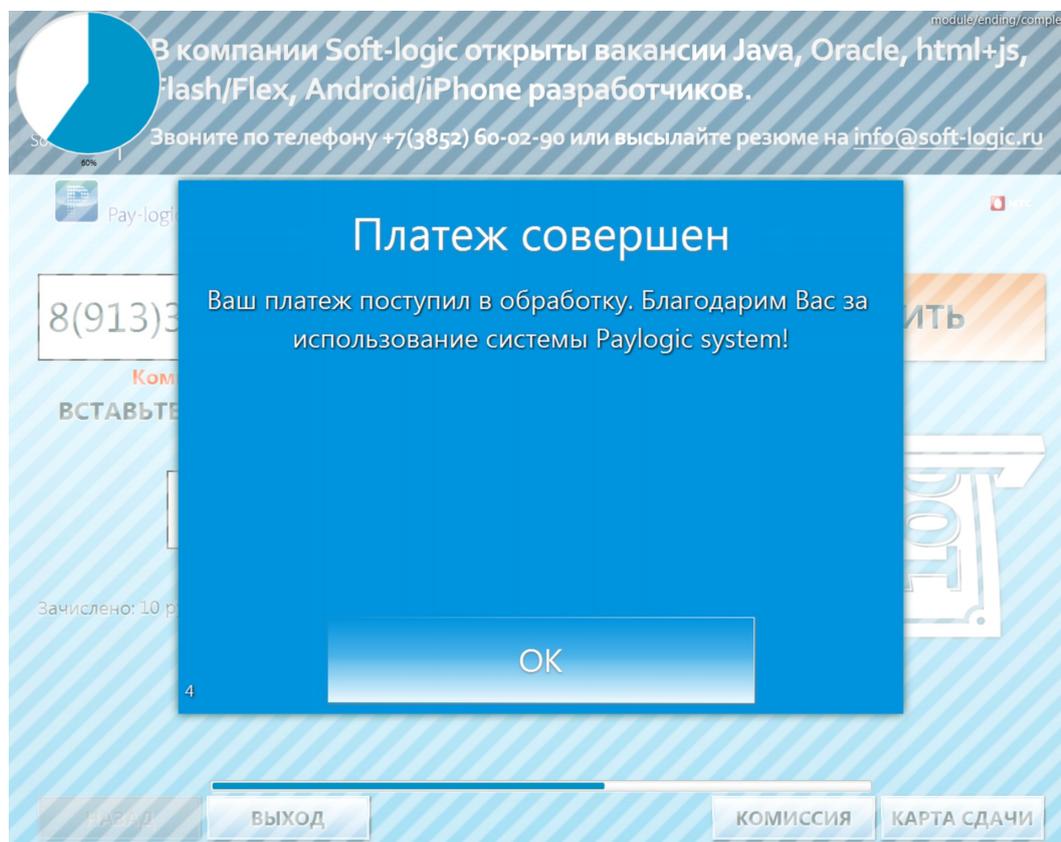


Рисунок 4.1.4.5 — Экран завершения платежа

#### 4.1.5 СПЕЦИАЛЬНЫЕ ЭКРАНЫ

К категории специальных экранов относятся все экраны, не участвующие в процессе оплаты и не являющиеся экранами меню, а выполняющие другие, в т. ч. сервисные функции.

**Экран поиска сервиса.** Позволяет найти сервис по фразе, введенной пользователем. Применяется для поиска нужной услуги среди всех услуг, которые есть на терминале. Показывает введенную фразу и кнопки сервисов, которые нашлись по указанным критериям, рисунок 4.1.5.1.

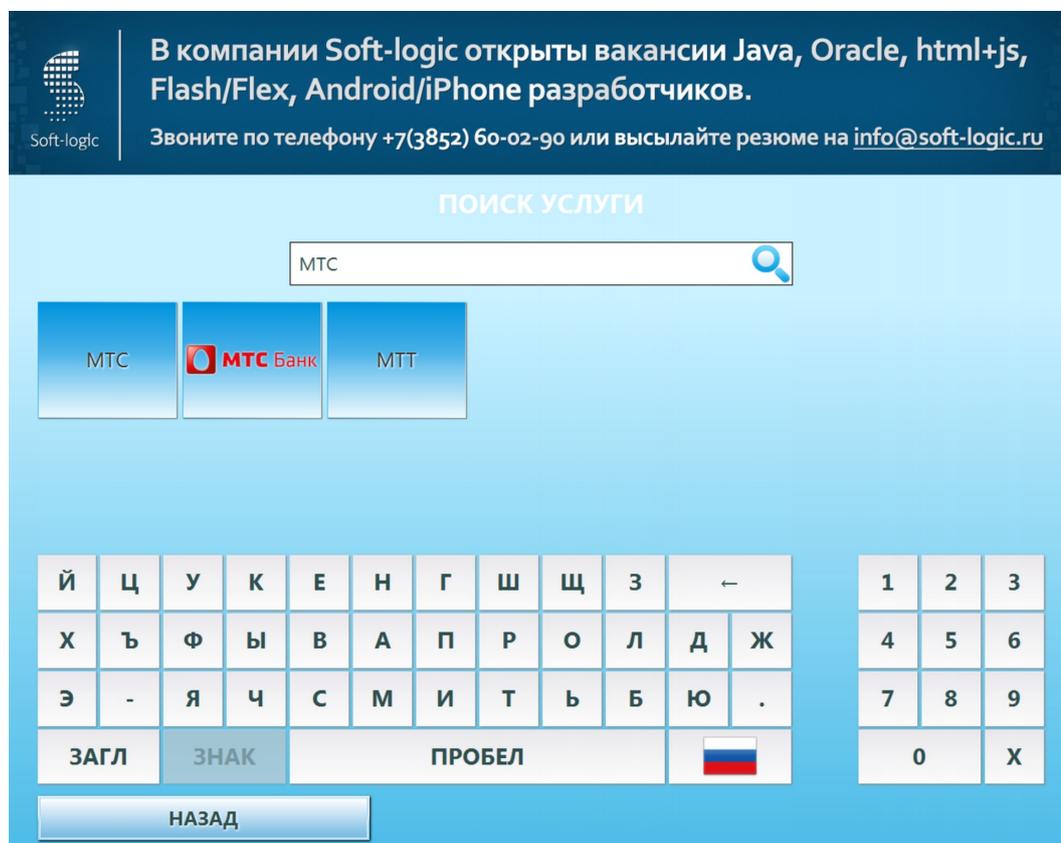
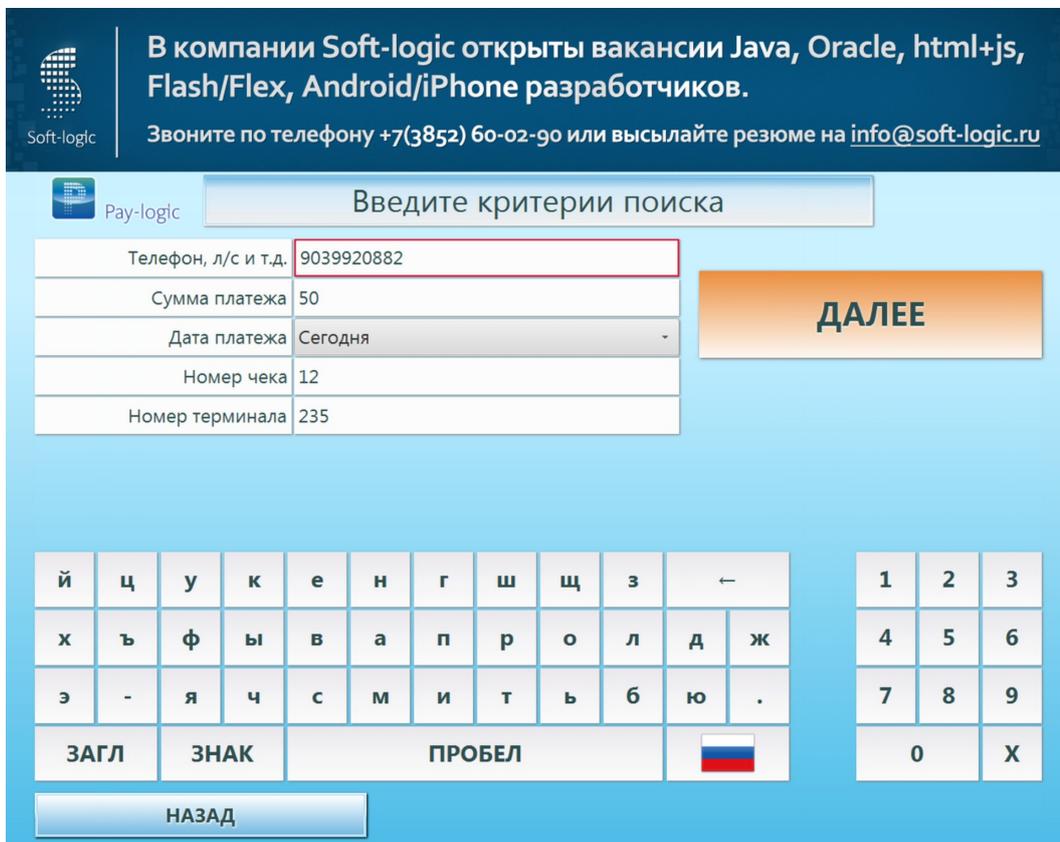


Рисунок 4.1.5.1 — Экран поиска сервиса

**Экран статуса платежа.** Служит для поиска платежа, ранее совершенного плательщиком. Позволяет, имея чек об оплате, проверить проведен ли платеж и его текущий статус в системе (рисунок 4.1.5.2).



В компании Soft-logic открыты вакансии Java, Oracle, html+js, Flash/Flex, Android/iPhone разработчиков.  
Звоните по телефону +7(3852) 60-02-90 или высылайте резюме на [info@soft-logic.ru](mailto:info@soft-logic.ru)

Pay-logic Введите критерии поиска

Телефон, л/с и т.д.	9039920882
Сумма платежа	50
Дата платежа	Сегодня
Номер чека	12
Номер терминала	235

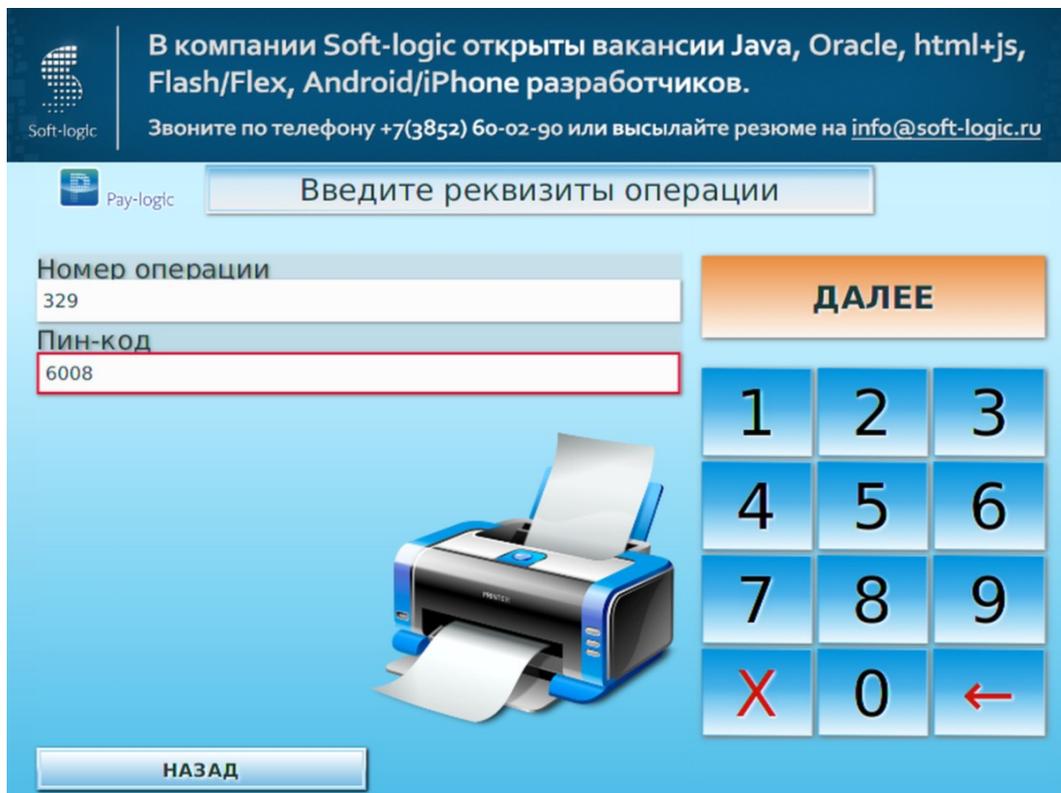
ДАЛЕЕ

й	ц	у	к	е	н	г	ш	щ	з	←	1	2	3	
х	ъ	ф	ы	в	а	п	р	о	л	д	ж	4	5	6
э	-	я	ч	с	м	и	т	ь	б	ю	.	7	8	9
ЗАГЛ	ЗНАК	ПРОБЕЛ								0	X			

НАЗАД

Рисунок 4.1.5.2 — Экран статуса платежа

**Экран копии чека.** Необходим для печати копии чека на терминале по пин-коду и номеру операции из процессинга (рисунок 4.1.5.3).



В компании Soft-logic открыты вакансии Java, Oracle, html+js, Flash/Flex, Android/iPhone разработчиков.  
Звоните по телефону +7(3852) 60-02-90 или высылайте резюме на [info@soft-logic.ru](mailto:info@soft-logic.ru)

Pay-logic Введите реквизиты операции

Номер операции  
329

Пин-код  
6008

ДАЛЕЕ

1 2 3  
4 5 6  
7 8 9  
X 0 ←

НАЗАД

Рисунок 4.1.5.3 — Экран копии чека

**Экран соглашения.** Служит для показа и просмотра соглашения (оферты). Обычно для форматирования применяются возможности языка HTML (рисунок 4.1.5.4).

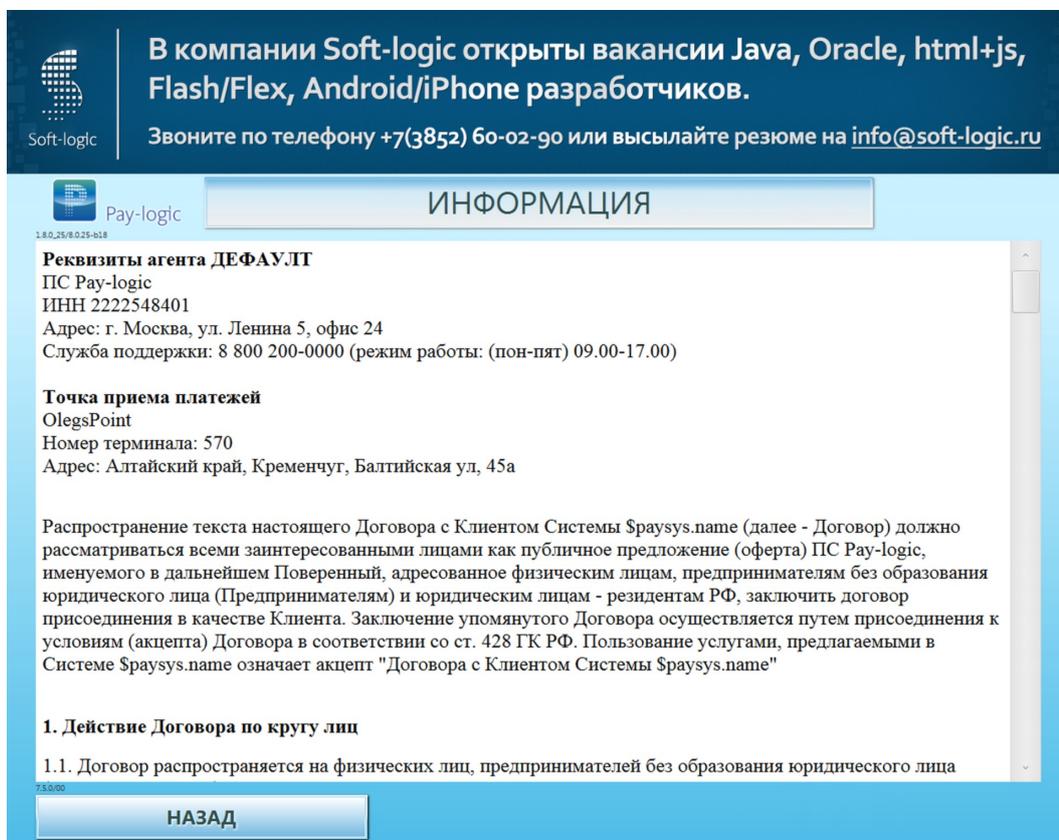


Рисунок 4.1.5.4 — Экран соглашения

**Экран помощи.** Служит для показа инструкции пользования терминалом. Также обычно для форматирования применяется HTML (рисунок 4.1.5.5).

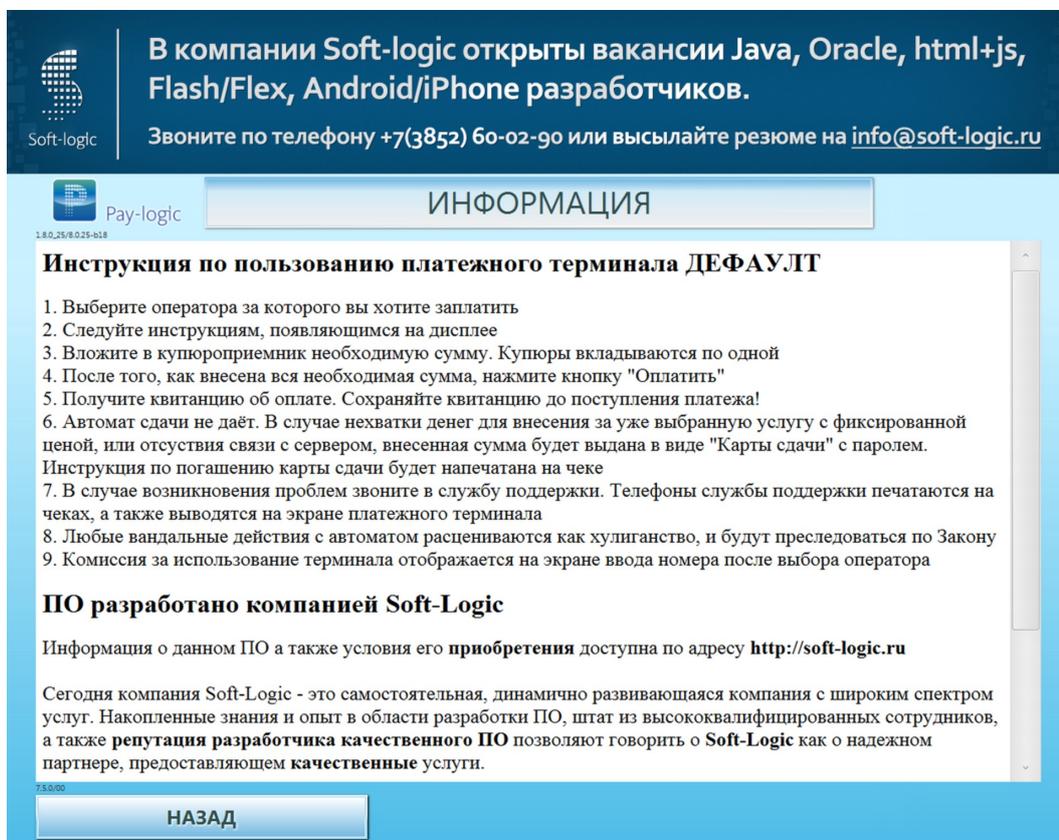


Рисунок 4.1.5.5 — Экран помощи

## 4.2 ТИПЫ ДИАЛОГОВ И ИХ СТРУКТУРА

Диалоговые окна — стандартные элементы интерфейса, обеспечивающие интерактивный вывод сообщений в различных ситуациях, выбор вариантов действий. Все они представляют собой модальные окна, при их показе основное ТПО затемняется. Разновидностью диалоговых окон в стандартном ТПО являются диалог выбора варианта сдачи и диалог завершения оплаты. Эти диалоги были рассмотрены выше в 4.1. Диалоги бывают различных видов, рассмотрим их последовательно.

**Однокнопочный (информационный).** Предполагает показ информации, закрытие его и возврат в основное окно. Применяется для информирования, в том числе вывода сообщений об ошибках (рисунок 4.2.1).

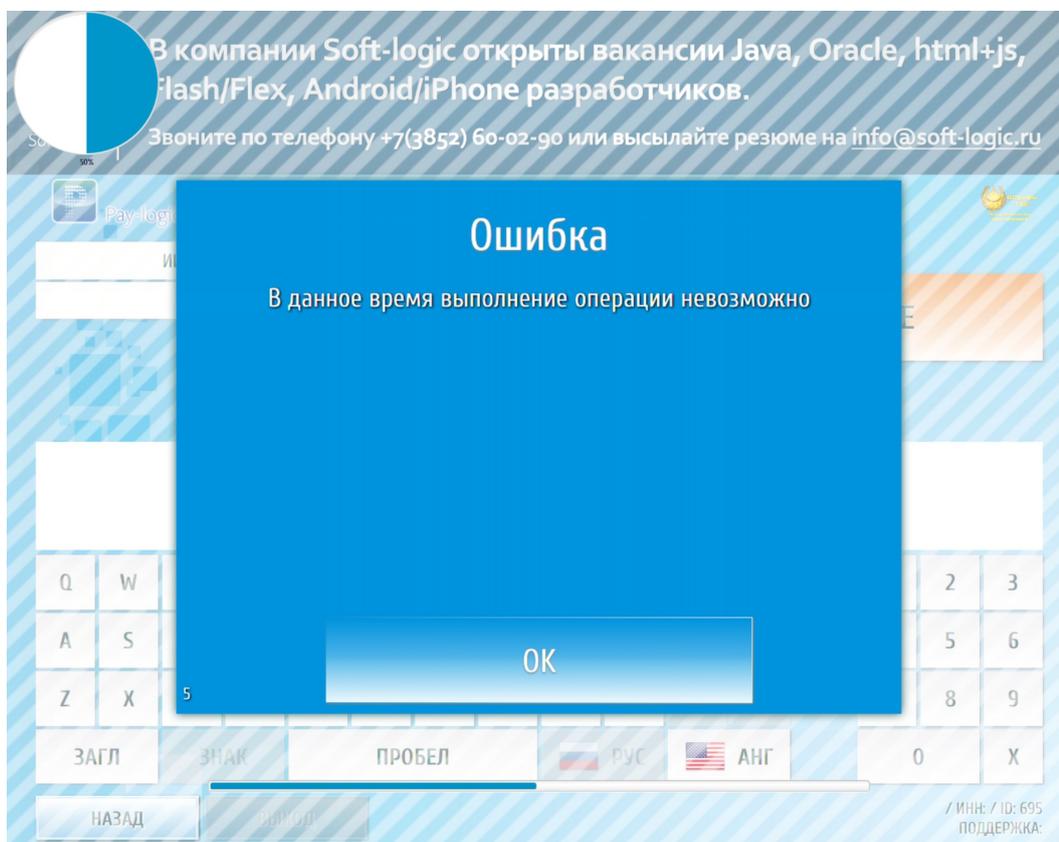


Рисунок 4.2.1 — Однокнопочный диалоговый экран

**Двухкнопочный диалог** — диалог, предполагающий вывод информации и последующий выбор одного из двух вариантов действий. Типовое применение — вывод информации и выбор продолжить действия или отменить действие либо выбор одного из двух дальнейших действий. Также может использоваться как информационный, но обычно предполагает выбор (рисунок 4.2.2).

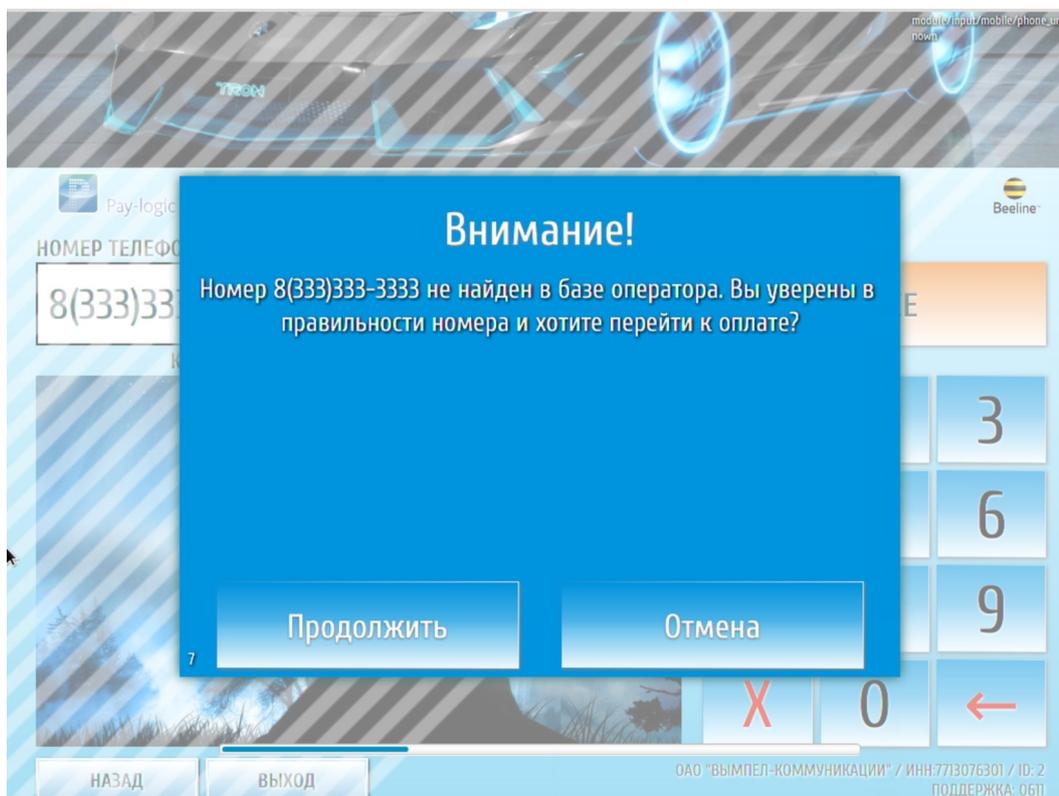


Рисунок 4.2.2 — Двухкнопочный диалоговый экран

**Трех- и более кнопочный диалог** — аналог двухкнопочного, но с большим числом вариантов действий. Содержит соответственно три или более кнопки, как правило, крайняя правая предполагает в таком случае отмену и возврат назад.

**Графический диалог** — диалоговое окно, в котором вариант выбора представлен иконкой или инфографикой. Может быть двух и более кнопочным. Пример такого диалога — разобранный выше диалог выдачи сдачи. По оформлению более привлекателен, но и более сложен в реализации (рисунок 4.2.3).

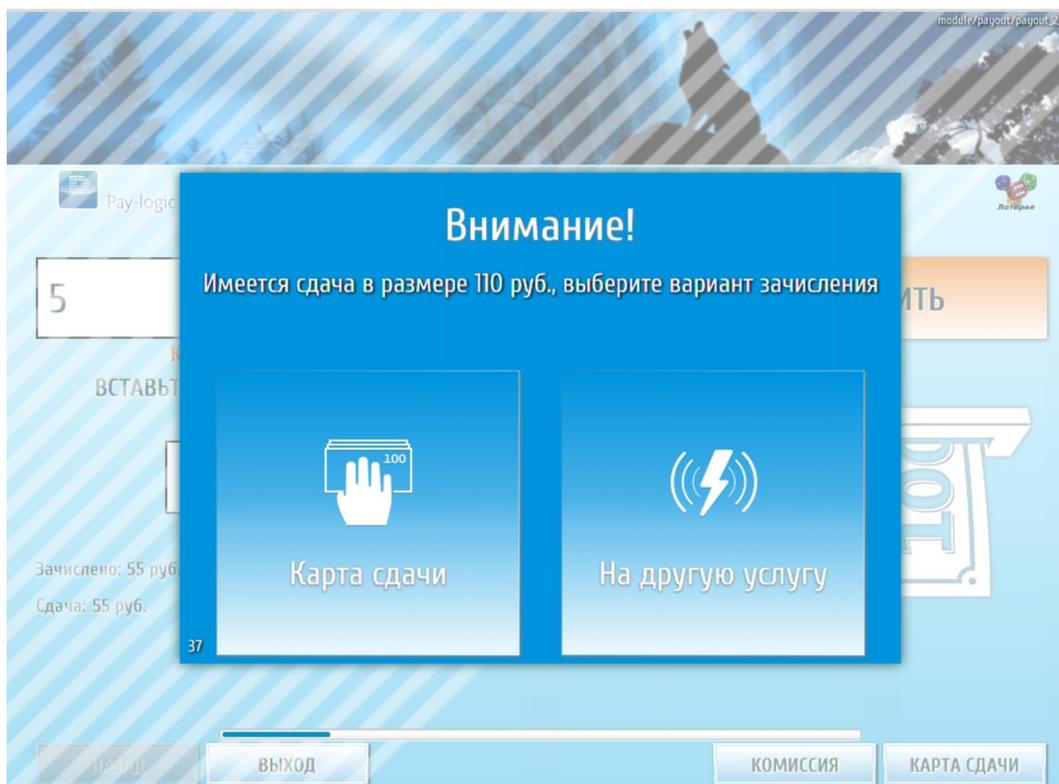


Рисунок 4.2.3 — Графический диалог

**Ресурсы диалога** — это текстовые сообщения и графические элементы, присутствующие в диалоге. Как правило, разработка дизайна требует отдельного выделения ресурсов в графические иконки. В примере выше ресурсы — это картинки вариантов и их названия. Текстовые ресурсы формируются непосредственно разработчиками, графические ресурсы должны быть предоставлены.

### 4.3 КНОПКИ

Кнопки — стандартные элементы интерфейса, предполагающие выполнение действия при нажатии на них. Речь идет о кнопках, имеющих самостоятельные функции в терминальном ПО и не входящих в состав диалогов, элементов ввода и клавиатур. Примерами кнопок могут быть пункты меню, кнопки перехода в разделы помощи, договора, поиска и т. д.

Выполняемые при нажатии на кнопки действия определяются контроллерами ТПО. Внешний вид кнопок определяется привязанными к кнопкам ресурсами (фоновая картинка или картинки, текстовка).

Кнопки меню разобраны в разделе экранов меню (раздел 4.1.2). С точки зрения дизайна этих кнопок важно, чтобы макет самого меню (категории сервисов, сервисы и горячие клавиши, их поведение) были дизайнером проработаны. То же самое относится ко всем остальным кнопкам, присутствующим на экране. Важно стилевое единство: если все кнопки на экранах (кроме меню) выполнены текстом, то этого следует придерживаться повсеместно, или наоборот, если принято решение добавить в кнопки иконки (инфографику) — она тоже должна быть повсеместно. Не рекомендуется смешивать подходы.

Кнопки (как и поля ввода и остальные элементы), как правило, типовые, одинакового или схожего размера и разных вариантов кнопок не так много. Дизайн должен содержать кнопки, отрисованные в состояниях: «не нажата», «нажата», «неактивна».

---

## 4.4 ЭЛЕМЕНТЫ ВВОДА

### 4.4.1 ОБЩИЕ СВЕДЕНИЯ

В терминальном ПО 7 версии применяется несколько типов элементов ввода. Каждый тип элемента ввода ориентирован на быстрый и удобный ввод данных определенного характера, максимально исключает ошибки при вводе данных. Элементы ввода являются составной частью экранов ввода данных, которые были разобраны выше. В этом разделе остановимся непосредственно на самих элементах ввода и их специфике. Элементами ввода являются:

1. Клавиатуры в различных вариантах.
2. Текстовые и числовые поля (включают клавиатуру).
3. Всплывающие клавиатуры.
4. Поля с автозаполнением (автокомплит).
5. Календарь как самостоятельный элемент.
6. Селекторы.
7. Корзина услуг.

Все экранные формы строятся на основе перечисленных элементов ввода.

---

#### 4.4.2 КЛАВИАТУРЫ

Клавиатура — необходимый элемент для того, чтобы пользователь имел возможность вводить данные. Терминальное ПО поддерживает 2 типа клавиатур:

1. Персистентная (обычная) клавиатура. Располагается на экране вместе с полем (полями) ввода и всегда на экране присутствует.
2. Всплывающая клавиатура. Не видна на экране до тех пор, пока пользователь не выберет поле для ввода, после чего появляется в отведенном для нее месте таким образом, чтобы не перекрывать выбранное поле.

По виду вводимых данных, необходимые клавиатуры бывают:

1. Алфавитно-цифровая — содержит буквы и цифры, возможность переключать язык ввода.
2. Цифровая — содержит только цифры и 2 символа помимо цифр: точку, запятую или другие (настраиваемо).
3. Полная клавиатура — содержит буквы, цифры, спец-символы.

Первые 2 вида клавиатур обязательны для ТПО, третий вид может отсутствовать или быть инкапсулирован в алфавитно-цифровую — на одной из раскладок показывать спецсимволы.

1. Алфавитно-цифровая клавиатура имеет вид, представленный на рисунке 4.4.2.1.

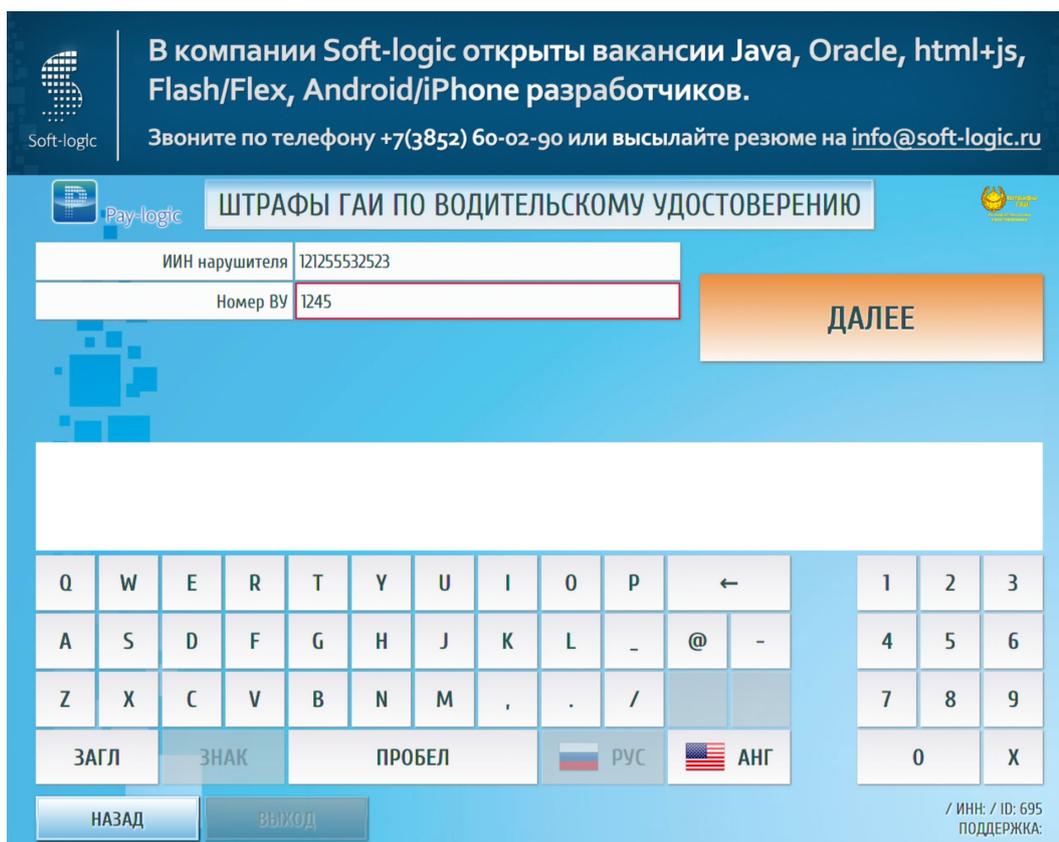


Рисунок 4.4.2.1 — Алфавитно-цифровая клавиатура

2. Цифровая клавиатура — рассмотрена выше.

С точки зрения разработки и создания клавиатур, важны:

1. Раскладка клавиатуры, т. е. последовательности символов на кнопках для цифр, букв, символов, букв разных языков из числа тех, что нужны заказчику.
2. Тип клавиатуры: простая или всплывающая.

Оба аспекта должны быть четко отражены в дизайне.

#### 4.4.3 ТЕКСТОВЫЕ И ЧИСЛОВЫЕ ПОЛЯ

Поля ввода не представляют ничего особенного. Для разработки важен проработанный дизайн этих полей. С точки зрения дизайна следует обращать внимание при компоновке экранов на следующее:

1. Вместимость полей — их ширина, размер и кегль шрифта должны обеспечивать безпроблемное вхождение среднестатистических значений. Шрифт не должен быть мелким, но и не должен быть слишком крупным.
2. Необходимость размещать на экранах рядом с полями информационных подсказок к этим полям. Места должно быть достаточно для ввода среднестатистической подсказки в одну строку, а сама подсказка не должна быть слишком мелкой.
3. Для экранов с одним полем (текстовое, числовое) — необходимо размещение справочной информации к полю (help), может быть объемным.
4. Для групповых экранов надо предусматривать размещение помощи (help) ко всему экрану.

#### 4.4.4 АВТОКОМПЛИТ

Элемент ввода предназначен для интерактивного поиска вариантов в процессе набора текста, позволяет вывести доступные варианты и дать пользователю выбрать требуемый. Автокомплиты работают с локальной базой данных и не требуют обращения к серверу. Возможна разработка автокомплитов обращающихся к серверу.

Визуально — это поле ввода, в которое вводится образец для поиска и предлагаются варианты, рисунок 4.4.4.1.



В компании Soft-logic открыты вакансии Java, Oracle, html+js, Flash/Flex, Android/iPhone разработчиков.

Звоните по телефону +7(3852) 60-02-90 или высылайте резюме на [info@soft-logic.ru](mailto:info@soft-logic.ru)


АЛСЕКО


Ввод данных
Выбор маршрута
Тип билета
Данные по студенческому

Пункт отправления	кемер	ДАЛЕЕ
Пункт прибытия	КАЗАХСТАН АККЕМЕР	
Дата:	КАЗАХСТАН КЕМЕР	
Дата:	ЛАТВИЯ КЕМЕРИ	
Введите пункт отправления	РОССИЯ КЕМЕРОВО РОССИЯ КЕМЕРОВО ПАСС РОССИЯ КЕМЕРОВО-СОТИРОВО.	

й	ц	у	к	е	н	г	ш	щ	з	←	
х	ъ	ф	ы	в	а	п	р	о	л	д	ж
э	-	я	ч	с	м	и	т	ь	б	ю	.
ЗАГЛ	ЗНАК	ПРОБЕЛ					РУС	АНГ			

1	2	3
4	5	6
7	8	9
0	X	

НАЗАД
ВЫХОД

/ ИНН: / ID: 746  
 ПОДДЕРЖКА:

Рисунок 4.4.4.1 — Автокомплит

## 4.5 ЭЛЕМЕНТЫ НАВИГАЦИИ

Навигация предназначена для перемещения пользователя между экранами ввода данных в процессе оплаты услуги. Навигация в форме кнопок «Назад» и «Далее» на экранах формы или сценария позволяет перейти на следующий или предыдущий шаг. Навигация в форме хлебных крошек позволяет вернуться сразу на любой предыдущий шаг, как правило, — чтобы исправить ошибку и продолжить выполнение сценария.

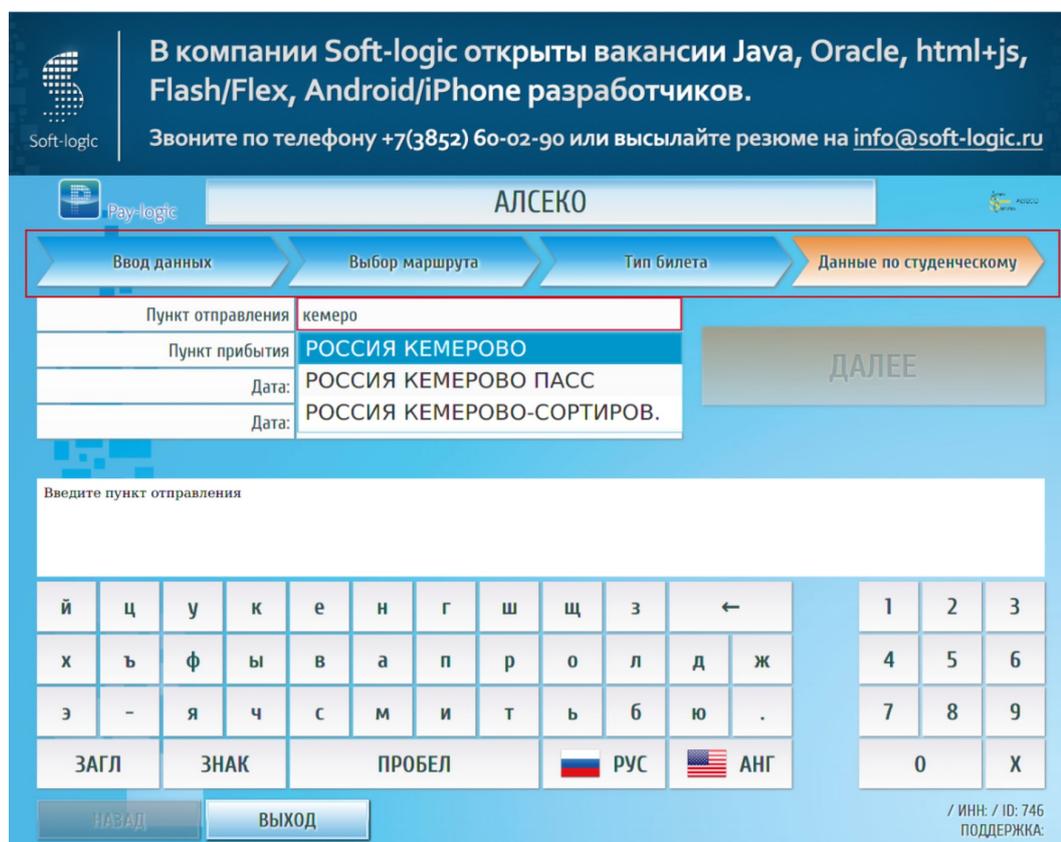


Рисунок 4.5.1 — Элементы навигации

Элементы навигации на рисунке 4.5.1 показаны красным выделением.

---

Кнопки внизу экрана «Назад» и «Выход» формально также относятся к элементам навигации, но в отличие от выделенных элементов пошаговой навигации позволяют сдвинуться только на один шаг сценария либо прервать его, в то время как полномаршрутная навигация позволяет вернуться на любой шаг назад.

---

## 4.6 РЕКЛАМА. РЕКЛАМНЫЕ МЕСТА И БАННЕРЫ

Рекламные места предназначены для организации показа рекламы на терминалах. Включают рекламные баннеры на главном экране, рекламные места на различных экранах терминального ПО.

Одним из рекламных мест может быть сквозной баннер. Как элемент всех экранов ТПО.

Баннерные места могут быть предусмотрены на любых экранах и в любом разумном количестве. При их размещении следует руководствоваться принципом удобства, чтобы баннеры не приводили к ухудшению удобства пользования терминалом, были не в ущерб и не мешали верно воспринимать информацию.

## 4.7 ТЕКСТОВКИ И ЛОКАЛИЗАЦИЯ

Текстовая информация на экранах представляет собой отдельные ресурсы. Это — названия экранов, заголовков, полей, кнопок навигации и даже букв на клавиатурах. Все текстовые элементы выносятся в ресурсные файлы:

1. Текстовки на экранах и кнопках (кроме относящихся к формам и сценариям) — в языковые ресурсные файлы.
2. Текстовки самих форм и сценариев — ведутся в сценариях и формах.
3. Названия услуг и групп меню — настраиваются на сервере.
4. Буквы на кнопках клавиатур — настраиваются в конфигурационном файле раскладок клавиатур.

С учетом мультиязычности все виды ресурсов представляются на всех языках. Технология работы следующая:

1. Определяется перечень языков.
2. Подготавливаются макеты раскладок клавиатур на всех языках.
3. Разрабатывается макет экранов на ОДНОМ языке, русском или английском.
4. Разработчики создают экраны.
5. Разработчики составляют ресурсные файлы, которые передаются клиенту для перевода на нужные языки.
6. После получения перевода — готовится мультиязычная сборка.

Не следует рисовать макеты на разных языках, это бесполезно и даже вредно. Также не следует предоставлять ресурсы и элементы клавиатур, содержащие на графике текст.

---

Так, например, для создания клавиатуры понадобится отрисованная кнопка в трех состояниях: нажато-отпущено-неактивно, но без какого-либо символа, нарисованного на ней — чистая кнопка. Символ будет размещен программно. То же самое относится ко всем остальным графическим ресурсам.

## 5 ТРЕБОВАНИЯ ДЛЯ РАЗРАБОТКИ ИНТЕРФЕЙСА

### 5.1 ЭТАПЫ СОЗДАНИЯ КАСТОМИЗИРОВАННОГО ИНТЕРФЕЙСА

Для успешной разработки кастомизированного под требования заказчика терминального ПО необходимо выполнить ряд шагов и представить разработчикам необходимый набор ресурсов. Основные этапы:

1. Определение требуемой функциональности и перечня необходимых экранов. Выполняется заказчиком.
2. Предоставление общего списка экранов в виде готовых прототипов или макетов. Выполняется заказчиком.
3. Согласование списка и функционала экранов, подготовка замечаний и дополнений. Выполняется сотрудниками компании «Soft-logic».
4. Подготовка рабочих макетов и ресурсов, необходимых для создания экранов. Выполняется Заказчиком.
5. Верстка экранов. Результатом верстки являются готовые экраны, скриншоты которых передаются на утверждение Заказчику. Выполняется сотрудниками компании «Soft-logic».
6. Реализация нестандартной бизнес-логики работы. Выполняется сотрудниками компании «Soft-logic».
7. Приемка Заказчиком первоначального варианта интерфейса.
8. Устранение замечаний и окончательная отладка интерфейса.

При подготовке информации для разработки экранов следует придерживаться рекомендаций, это позволит существенно сократить срок и повысить качество.

---

## 5.2 СОГЛАСОВАНИЕ ПЕРЕЧНЯ ЭКРАНОВ

Для согласования необходимо предоставить перечень всех экранов ТПО, которые понадобятся клиенту. Перечень может быть оформлен в любом удобном для клиента виде. Специальных требований к формату нет, но перечень обязательно должен давать представление о функциональности будущего терминального ПО и о его внешнем виде, содержать для каждого экрана: название, функциональное назначение, описание и общий вид экрана.

**Название** необходимо для идентификации экрана, нужно для того, чтобы при детальном обсуждении экрана на него ссылаться и исключить путаницу. Следует придерживаться правила: название экрана должно соответствовать типу одного из стандартных экранов либо дополнять/расширять один из стандартных типов экрана.

**Назначение экрана** — это описание выполняемой экраном функции: для какой бизнес-цели экран предназначен, какую роль в терминальном ПО выполняет. Указывается всегда, даже если роль представляется очевидной.

**Описание экрана** — это либо краткое (если нет неочевидных особенностей), либо развернутое (если экран реализует сложную, либо сильно отличающуюся от стандартной для аналогичного экрана «Soft-logic» бизнес-логику). Чем подробнее изложена логика работы экрана, тем лучше.

**Общий вид экрана** — это либо схематичный макет, либо полноценно сверстаный дизайнером готовый экран, позволяющее понять, как именно экран должен выглядеть, примерное расположение элементов интерфейса и т. д.

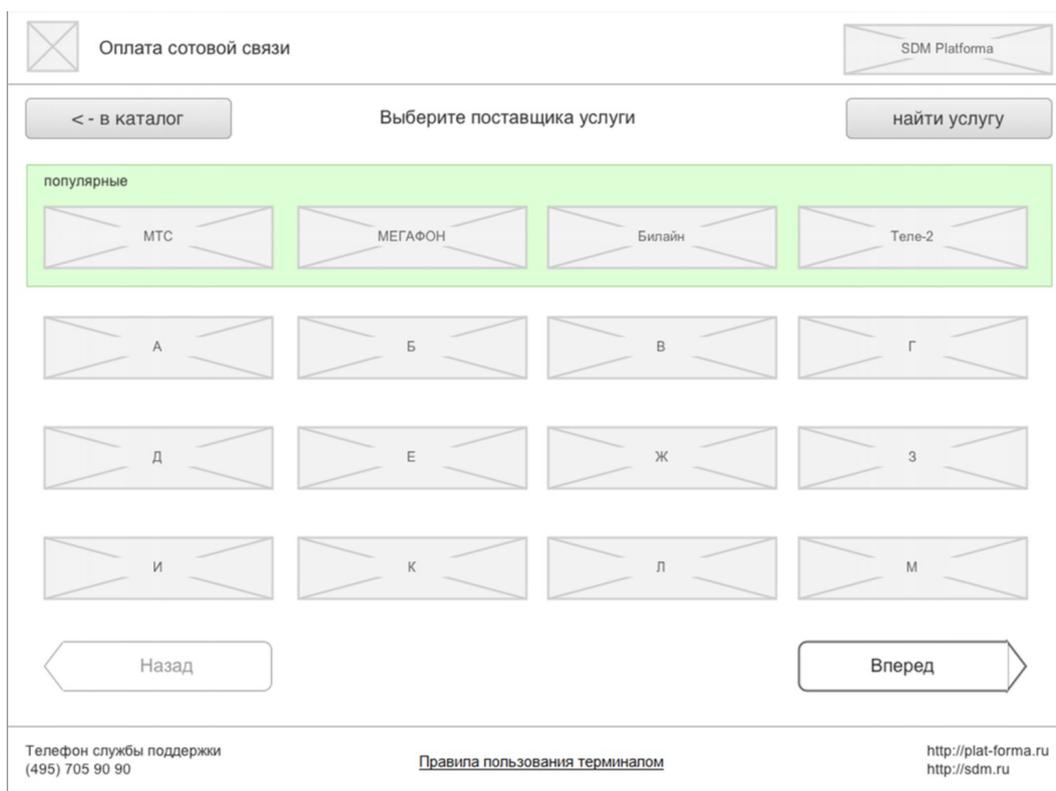


Рисунок 5.2.1 — Пример общего вида экрана: макет



Рисунок 5.2.2 — Пример общего вида экрана: готовый экран

Для экранов, логика работы которых не отличается от стандартных экранов компании «Soft-logic» вид или макет, тем не менее, необходим, так как внешние отличия есть всегда.

### Пример:

**Название:** Экран ввода цифровых значений

**Назначение:** Ввод цифровых данных, таких как номера телефонов, договоров, суммы платежей

**Описание:** Сверху экрана присутствует заголовок, на котором отображается логотип платежной системы, название сервиса и иконка сервиса. В центре экрана располагаются одно поле с заголовком и цифровая клавиатура, при помощи которой и осуществляется ввод. Клавиатура помимо цифровых клавиш имеет две управляющих: «очистить» (расположена слева) — полностью очищает поле ввода, «удалить последний» (расположена справа) — удалить последнюю цифру. Имеются три кнопки навигации: «Назад» — вернуться на предыдущий экран, «Выход» — выйти в главное меню, «Далее» — перейти к следующему экрану. Кнопка «Далее» должна быть неактивной до тех пор, пока пользователь не ввел верные данные. Внизу экрана присутствует надпись, информирующая о номер терминала и номере технической поддержки.

Или: **Описание:** Соответствует стандартному экрану ввода цифровых данных в Soft-logic

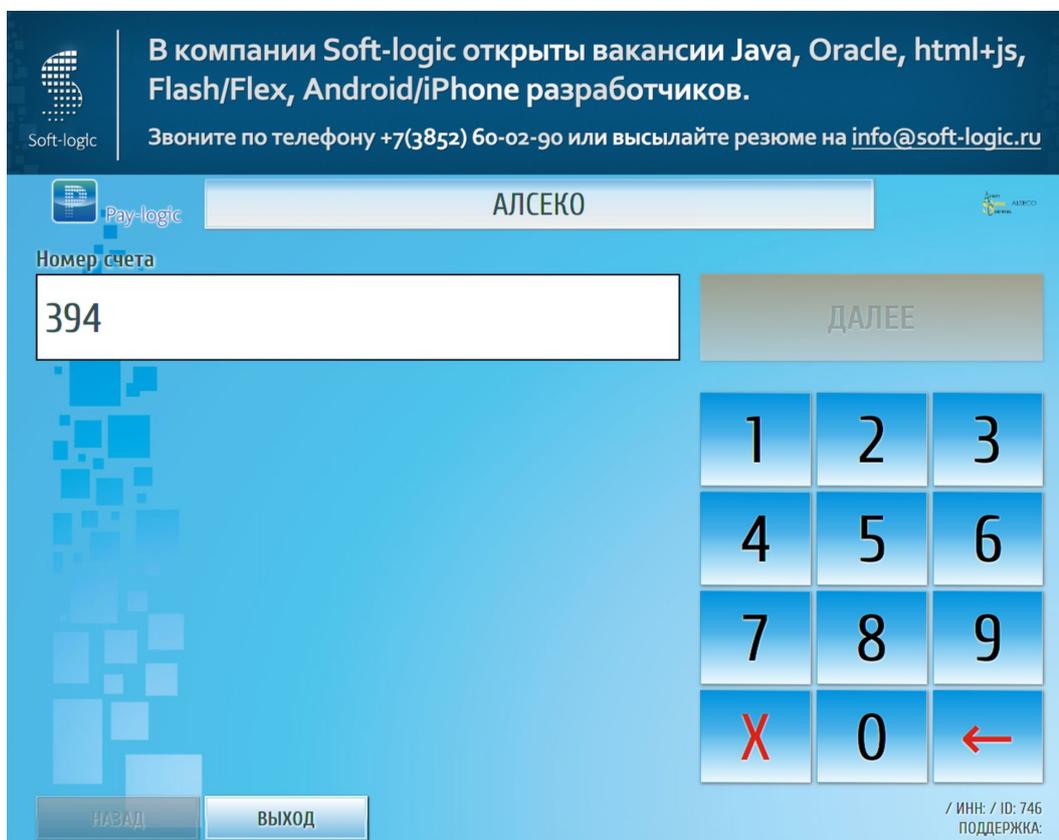


Рисунок 5.2.3 — Общий вид

---

Список экранов изучается специалистами и при необходимости дополняется. По готовности списка экранов Заказчику необходимо подготовить рабочие макеты и ресурсы.

### 5.3 ПОДГОТОВКА РАБОЧИХ МАКЕТОВ И РЕСУРСОВ

**Рабочий макет** — окончательный вариант макета экрана, содержащий всю необходимую информацию о расположении элементов относительно друг друга, позволяющий точно воспроизвести экран в процессе верстки.

**Ресурсы** — элементы экрана, необходимые в виде отдельных файлов (картинок, шрифтов).

**Стили** — описание элементов, которые потребуются для верстки элементов средствами стилей CSS.

Для каждого экрана в отдельности необходимо предоставить эталонный макет. Есть несколько вариантов представления эталонного макета экрана.

1. **Раскладка по сетке.** Представляет из себя макет экрана с нанесенной сеткой заданного размера (например, 32 пикселя), рисунок 5.3.1.

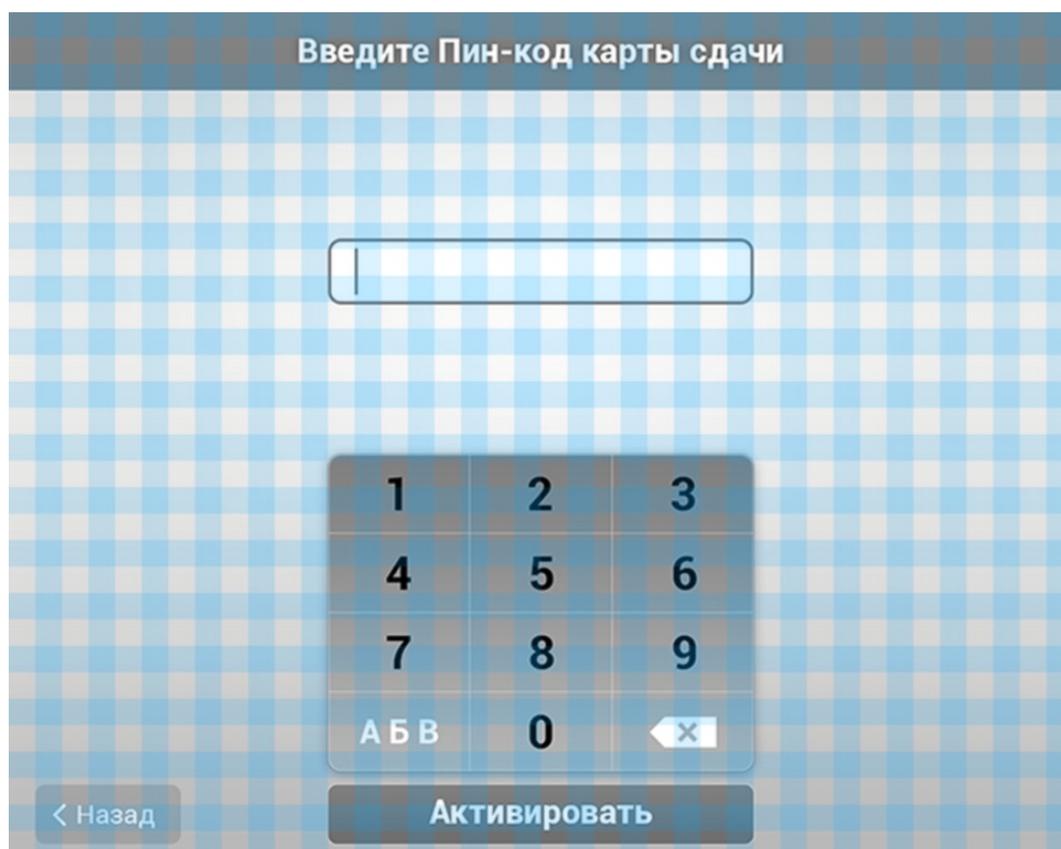


Рисунок 5.3.1 — Раскладка по сетке

**2. Размерные линии (выноски).** Представляет из себя макет экрана с указанием отступов от краев экрана, элементов друг от друга — в пикселях, рисунок 5.3.2.

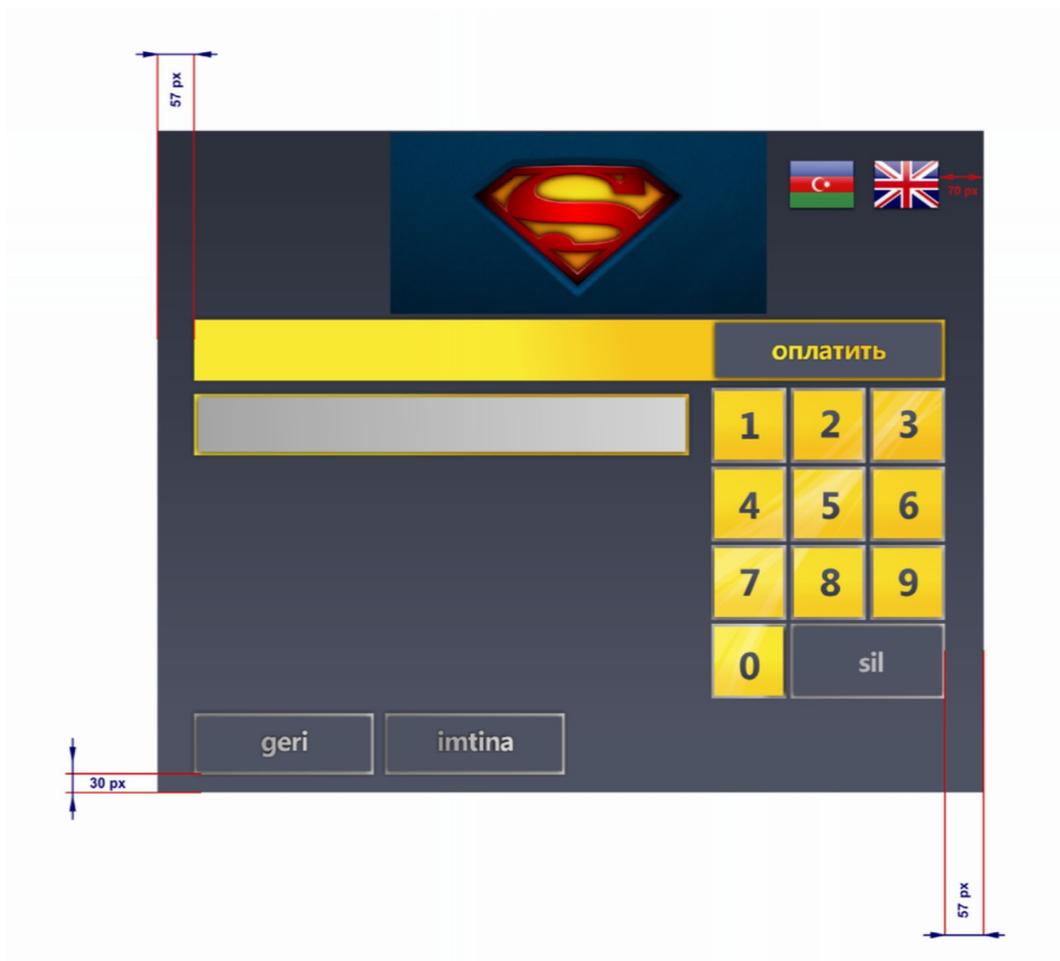


Рисунок 5.3.2 — Размерные линии

Обязательно в дополнение к эталонному макету предоставить чистый вид готового экрана, т.е. полностью готовое чистовое изображение экрана в полномразмерном разрешении и без любых дополнительных линий. Чистый вид применяется разработчиком для визуальной сверки сверстанного экрана с нужным итоговым результатом.

**Стили** могут быть предоставлены следующим образом:

1. **Таблица стилей** — Заказчиком заполняется таблица с описанием стилей для каждого элемента экрана. Наиболее сложный для Заказчика, но и наиболее

быстрый для реализации со стороны разработки вариант. Заполняется таблица следующего содержания (таблица 5.3.1, 5.3.2).

Таблица 5.3.1 — Таблица стилей

Элементы интерфейса	Ширина, высота	Цвет, непрозрачность	Градиент	Цвет обводки, толщина, непрозрачность	Радиус скругления	Цвет тени, смещение, размытие, непрозрачность	Текст в элементе: шрифт, размер, цвет	Цвет тени текста, смещение, размытие, непрозрачность
Кнопка навигации	128px 96px	#dc3a4d, 100%		#ffffff, 4px, 50%	12px		Roboto Bold, 40px, #ffffff	
Поле ввода	1024 x 80px		#cadce6, ecf8ff 90град		15зч	#005785, 2px, 16px, 40%	Roboto Regular, 28px, #8b6019	

Таблица 5.3.2 — Таблица стилей. Продолжение

Текст	Шрифт	Размер	Цвет
Заголовок на банере	Roboto Regular	40px	#334a55
Заголовок панели	Roboto Regular	56px	#334a55

**Стили должны быть заполнены для абсолютно всех элементов, присутствующих на экране.** Так как на многих экранах стили дублируются, допускается ссылаться на другие экраны, указывая, что для элемента применен стиль такой же, как и, например, на экране оплаты.

**2. Photoshop** — передача разработчикам исходника оригинал-макета в формате Adobe Photoshop (\*.PSD). В таком случае таблица стилей не заполняется, а всю необходимую информацию об элементе разработчик извлекает из структуры оригинал-макета.

**Внимание!**

Необходимо, чтобы оригинал-макет был построен со слоями, а сами слои должны быть сгруппированы по элементам так, чтобы была возможность извлечь как отдельный элемент целиком (кнопку, поле и т.д), так и ее составляющие: контур, подложку, надпись (шрифт и текст), радиусы закруглений, градиенты и т. д. Сведенные слои не позволят выполнить эту работу и потому не принимаются.

Компания «Soft-logic» оставляет за собой право в случае трудностей с извлечением стилей из оригинал-макета Adobe Photoshop обращаться за разъяснениями и ответами на вопросы к Заказчику, поскольку иногда дизайнерами применяются нестандартные приемы работы со слоями, элементами и пр. В случае, если возникли трудности такого рода мы просим предоставить стили одного конкретного элемента экрана, который такие трудности вызвал.

**Ресурсы.** Включают в себя иконки (картинки) и шрифты.

**Картинки (иконки)** представляют собой графические изображения, которые нельзя отрисовать стилями CSS - JavaFx UI (сложные кривые, сложные графические элементы, кнопки сложной формы и т.д). Картинки (иконки) предоставляются для каждого экрана в формате \*.PNG в масштабе 100%. Если разработчикам предоставляется оригинал-макет в формате Adobe Photoshop, дополнительно прикладывать отдельные картинки не требуется.

**Шрифты.** Необходимо приложить шрифты, которые использовал дизайнер при верстке макетов. Шрифты прикладываются в формате .ttf. При выборе шрифта очень важно учитывать, что многие шрифты в разных операционных системах отрисовываются по-разному, поэтому рекомендуем заранее проверять, что выбранный вами шрифт одинаково (примерно одинаково) отрисовывается на системах семейства Windows и Linux.

Рабочие макеты необходимо предоставлять в виде одного архива, внутри архива для каждого экрана должна быть отдельная папка, в которой размещено всё необходимое,

---

относящееся к этому экрану. Название папки должно соответствовать названию экрана согласно разделу [5.2](#).