Министерство образования Республики Беларусь

Учреждение образования  
Белорусский государственный университет  
информатики и радиоэлектроники

Факультет Компьютерных систем и сетей

Кафедра Информатики

*К защите допустить:*

Заведующий кафедрой информатики

\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.А. Волорова

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту

на тему:

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ**

**КЛИЕНТСКОГО ТЕРМИНАЛА**

**ДЛЯ ОБМЕНА ВАЛЮТ**

БГУИР ДП 1‑40 01 03 00 019 ПЗ

Студент: Е.В. Евстратенко

Руководитель: В.С. Иванов

Консультанты:

*от кафедры информатики* В.В. Шиманский

*по экономической части* К.Р. Литвинович

Нормоконтролер В.В. Шиманский

Рецензент:

Минск 2016

**РЕФЕРАТ**

СИСТЕМА САМООБСЛУЖИВАНИЯ, ОБМЕН ВАЛЮТ, ПРОЕКТИРОВАНИЕ, СИСТЕМА, СЕРВЕР, КЛИЕНТ, БД.

Дипломный проект выполнен на 6 листах формата А1 с пояснительной

запиской на 79 страницах без приложений справочного или информационного характера. Пояснительная записка включает 8 разделов, 24 рисунков, 9 таблиц, 40 формул и 6 литературных источников.

Объектом разработки является программное обеспечение клиентского терминала для обмена валют.

Целью дипломного проекта является проектирование и реализация системы, пригодной для проведения валютно-обменных операций.

Основное назначение комплекса – это частично автоматизировать рабочее место сотрудника банка. Такая система при высоком уровне реализации вполне способна облегчить и упростить валютно-обменный процесс, а также сэкономить время как для банка, так и для клиентов, за счет минимизации временных затрат, связанных с рутинными операциями.

Для достижения цели дипломного проекта было выполнено логическое и физическое моделирование данных, спроектированы и реализованы база данных, серверная и клиентские части программного комплекса, предназначенные для использования и администрирования системы.

В разделе технико-экономического обоснования был произведен расчет затрат на создание ПО, а также прибыль от внедрения, получаемая организацией. Произведенные расчеты показали экономическую целесообразность проекта.

**СОДЕРЖАНИЕ**

[ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ И ТЕРМИНОВ 6](#_Toc451994224)

[ВВЕДЕНИЕ 7](#_Toc451994225)

[1 ПОСТАНОВКА ЗАДАЧИ 8](#_Toc451994226)

[2 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ 10](#_Toc451994227)

[2.1 Устройства банковского самообслуживания 10](#_Toc451994228)

[2.1.1 Банкоматы 10](#_Toc451994229)

[2.1.2 Платежные терминалы 10](#_Toc451994230)

[2.1.3 Информационные киоски 11](#_Toc451994231)

[2.2 Средства разработки 11](#_Toc451994232)

[2.2.1 WPF 11](#_Toc451994233)

[2.2.2 WCF 12](#_Toc451994234)

[2.2.3 ASP.NET MVC 14](#_Toc451994235)

[2.2.4 Microsoft SQL Server 15](#_Toc451994236)

[3 ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К СИСТЕМЕ 17](#_Toc451994237)

[3.1 Требования к функциональным характеристикам системы 17](#_Toc451994238)

[3.2 Требования к составу и параметрам технических средств ВОТ 17](#_Toc451994239)

[3.3 Требования к ПС терминала 18](#_Toc451994240)

[3.4 Требования к надежности 19](#_Toc451994241)

[4 ПРОЕКТИРОВАНИЕ СИСТЕМЫ 21](#_Toc451994242)

[4.1 Выбор методологий моделирования и инструментария 21](#_Toc451994243)

[4.2 Разработка диаграмм вариантов использования 22](#_Toc451994244)

[4.2.1 Действующие лица 23](#_Toc451994245)

[4.2.2 Варианты использования 23](#_Toc451994246)

[4.2.3 Диаграмма вариантов использования 23](#_Toc451994247)

[4.2.4 Описание вариантов использования 24](#_Toc451994248)

[4.3 Построение логической модели данных 27](#_Toc451994249)

[5 ФИЗИЧЕСКОЕ МОДЕЛИРОВАНИЕ 29](#_Toc451994250)

[5.1 Физическая модель данных 29](#_Toc451994251)

[5.1.1 Проектирование и создание таблиц базы данных 29](#_Toc451994252)

[5.2 Компоненты предмета разработки 30](#_Toc451994253)

[5.3 Описание компоне-нтов ПС терминала 31](#_Toc451994254)

[5.3.1 ПК купюроприемника 31](#_Toc451994255)

[5.3.2 ПК диспенсера 32](#_Toc451994256)

[5.3.3 ПК системы видеонаблюдения 33](#_Toc451994257)

[5.3.4 ПК чекового принтера 34](#_Toc451994258)

[5.3.5 ПК блока датчиков 35](#_Toc451994259)

[5.3.6 ПК источника бесперебойного питания 35](#_Toc451994260)

[5.3.7 Управляющая программа ВОТ 36](#_Toc451994261)

[5.4 Развертывание предмета разработки 37](#_Toc451994262)

[6 РЕАЛИЗАЦИЯ СИСТЕМЫ 39](#_Toc451994263)

[6.1 Назначение и описание компонентов ПО 39](#_Toc451994264)

[6.2 Режимы работы ВОТ 44](#_Toc451994265)

[7 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 47](#_Toc451994266)

[7.1 Работа с ПС терминала 48](#_Toc451994267)

[7.1.1 Ожидание пользователя 48](#_Toc451994268)

[7.1.2 Информация по транзакции 49](#_Toc451994269)

[7.1.3 Вход в сервисный режим 50](#_Toc451994270)

[7.1.4 Вкладка «Инкассация» 51](#_Toc451994271)

[7.1.5 Вкладка «Оборудование» 54](#_Toc451994272)

[7.1.6 Выход из сервисного режима 55](#_Toc451994273)

[7.2 Работа с ПС бизнес администратора 55](#_Toc451994274)

[7.2.1 Аутентификация 55](#_Toc451994275)

[7.2.2 Отображение списка терминалов 56](#_Toc451994276)

[7.2.3 Добавление нового терминала 56](#_Toc451994277)

[7.2.4 Просмотр детальной информации терминала 57](#_Toc451994278)

[7.2.5 Установка курсов валют 58](#_Toc451994279)

[7.2.6 Установка принимаемых банкнот 60](#_Toc451994280)

[8 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ДИПЛОМНОГО ПРОЕКТА 61](#_Toc451994281)

[8.1 Введение и исходные данные 61](#_Toc451994282)

[8.2 Расчет сметы затрат и цены программного продукта 61](#_Toc451994283)

[8.3 Расчет нормативной трудоемкости 63](#_Toc451994284)

[8.4 Расчет основной заработной платы исполнителей 67](#_Toc451994285)

[8.5 Оценка экономической эффективности применения ПП у пользователя 71](#_Toc451994286)

[8.6 Расчет капитальных затрат 74](#_Toc451994287)

[8.7 Расчет экономического эффекта 75](#_Toc451994288)

[8.8 Выводы по технико-экономическому обоснованию 76](#_Toc451994289)

[ЗАКЛЮЧЕНИЕ 78](#_Toc451994290)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 80](#_Toc451994291)

# 

# ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ И ТЕРМИНОВ

АРМ – автоматизированное рабочее место

БД – база данных

ВОТ – валютно-обменный терминал

ИБП, UPS – источник бесперебойного питания

ОС – операционная система

ПК – программный компонент

ПО – программное обеспечение

ПС – программное средство

ССВОО – система самообслуживания валютно-обменных операций

СУБД – система управления базами данных

# ВВЕДЕНИЕ

Банковский сектор в Беларуси один из самых динамичных. Постоянное развитие банковского сектора приводит к ужесточению конкуренции, вследствие чего ставится вопрос улучшения качества обслуживания.

Для улучшения качества обслуживания банковские учреждения обращаются к стратегии, направленной на создание зон самообслуживания с использованием машин нового поколения — банкоматов, инфокиосков, терминалов.

Устройства самообслуживания позволяют клиенту выполнять различные банковские операции, не привлекая для этого банковских работников. Спрос на устройства самообслуживания в банковском секторе растет очень быстрыми темпами.

Устанавливая в своих отделениях Cash-In банкоматы, банки решают целый ряд насущных проблем. Во-первых, это скорость обслуживания клиентов. Немаловажно и то, что банки сокращают затраты на проведение транзакций. Используя Cash-In банкоматы, отделение банка может ощутимо снижать операционные и административные издержки. Но и это далеко не все преимущества. С внедрением Cash-In персонал освобождается от выполнения рутинных кассовых операций, более того, банкомат даже лучше справляется с такими задачами. При этом выиграют и клиент, и банк, который при помощи таких киосков существенно разгружает кассиров, освобождая их для более сложной, высококвалифицированной работы.

Для потребителя отсутствует необходимость стоять в очереди в кассу, банкоматы дают возможность пользоваться банковскими услугами в режиме 7 на 24, что особенно важно для сегмента работающих клиентов, которые не могут посетить банковское отделение в рабочие часы. В центре самообслуживания клиент может получить исчерпывающую информацию о работе и услугах банка. Также преимуществом для клиента является то, что тарифы на услуги в залах самообслуживания существенно ниже, чем на аналогичные услуги, предоставляемые кассирами и менеджерами отделений.

В данном дипломном проекте рассматривается практика создания программного обеспечения клиентского терминала для обмена валют. В пояснительной записке рассмотрен полный цикл разработки программного обеспечения.

# ПОСТАНОВКА ЗАДАЧИ

Целью дипломного проекта является разработка программного обеспечения клиентского терминала для обмена валют.

Основное назначение терминала – автоматизировать процесс обмена валют, а также создать удобный графический интерфейс для работы с терминалом как клиентов, так и обслуживающего персонала.

В рамках реализации данного проекта должны быть разработаны следующие программные средства:

– программное средство терминала (далее ПС терминала);

– программное средство сервера (далее ПС сервера);

– программное средство бизнес администратора (далее ПС бизнес администратора.

*ПС терминала* будет установлено на каждом валютно-обменном терминале (далее ВОТ) и предоставит следующие функциональные возможности:

1. обмен иностранных купюр (доллары США, евро, российские рубли) на белорусские рубли по установленным для обмена курсам;
2. предоставление необходимой справочной информации: курсы валют, список принимаемых банкнот;
3. предоставление рекламной информации;
4. аутентификация и проведение инкассации;
5. отображение информации о состоянии терминала в режиме инкассации;
6. выбор и установка языковой версии для проведения операции обмена. Для выбора будут доступны следующие языковые версии:

– русская;

– белорусская;

– английская;

– немецкая.

Работа с терминалом должна быть максимально простой и удобной. Все действия и операции, выполняемые на терминале, должны быть интуитивно понятны. Для разработки приложения должна использоваться технология WPF.

*ПС сервера* будет опубликовано в локальной сети и позволит:

1. передавать курсы обмена валют каждому ВОТ системы;
2. передавать каждому ВОТ системы купюры, принимаемые ВОТ к обмену;
3. передавать ВОТ информацию, необходимую для обмена купюры;
4. для каждого ВОТ системы вести учет количества купюр в кассетах;
5. формировать отчетные данные об операциях, совершенных на каждом ВОТ системы;
6. выполнять круглосуточный мониторинг состояния ВОТ системы.

Для разработки серверного приложения выбрана технология WCF. Приоритетными требованиями являются скорость и надежность.

*ПС бизнес администратора* также будет опубликовано в локальной сети. Перед началом работы необходима аутентификация пользователя.

ПС бизнес администратора позволит:

1. аутентифицировать пользователя;
2. для каждого ВОТ системы устанавливать курсы обмена валют;
3. для каждого ВОТ системы устанавливать номиналы купюр в разрезе валют принимаемые к обмену;
4. по каждому ВОТ системы отслеживать количество купюр различных номиналов в кассетах его купюроприемника;
5. запрашивать отчетные данные об операциях, совершенных на каждом ВОТ системы;
6. отображать информацию по всем ВОТ системы в графических интерфейсах ПС бизнес администратора.

Технологией разработки веб-приложения выбрана технология ASP.NET MVC. Обеспечение безопасности является ключевым требованием и достигается за счет механизма авторизации, а также разделения прав и ролей пользователей.

Все приложения должны быть разработаны под семейство операционных систем Windows в интегрированной среде разработки Visual Studio 2013 и выше с использованием платформы .NET на базе Framework 4.5 и выше и языка программирования C#, а также технологии доступа к данным EF. В качестве СУБД должен использоваться MSSQL Server 2014.

# ТЕОРЕТИЧЕСКИЕ ОСНОВЫ

## Устройства банковского самообслуживания

Технологии дистанционного банковского обслуживания (ДБО) с использованием устройств банковского самообслуживания являются одними из наиболее популярных в мире.

Банкоматы и терминалы попадают в категорию ДБО, так как почти полностью предоставляют банковские услуги дистанционно, без посещения клиентом банковской организации. Кроме того, важным фактором для включения их в эту категорию является возможность дублирования основных функций стандартного банк-клиента, который банк предоставляет частным (физическим) лицам для осуществления платежей.

Можно выделить несколько видов ДБО по типу используемых устройств:

1. ДБО с использованием банкоматов;
2. ДБО с использованием [платежных терминалов](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D1%82%D1%91%D0%B6%D0%BD%D1%8B%D0%B9_%D1%82%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%B0%D0%BB);
3. ДБО с использованием [информационных киосков](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BA%D0%B8%D0%BE%D1%81%D0%BA).

### Банкоматы

Банкомат – программно-технический комплекс, предназначенный для автоматизированных выдачи и приема наличных денежных средств как с использованием [платежных карт](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D1%82%D1%91%D0%B6%D0%BD%D0%B0%D1%8F_%D0%BA%D0%B0%D1%80%D1%82%D0%B0), так и без, а также выполнения других операций, в том числе оплаты товаров и услуг, составления документов, подтверждающих соответствующие операции.

### Платежные терминалы

Платежный терминал – [аппаратно-программный комплекс](https://ru.wikipedia.org/wiki/%D0%90%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D0%BE-%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%BC%D0%BF%D0%BB%D0%B5%D0%BA%D1%81), обеспечивающий прием [платежей](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D1%82%D1%91%D0%B6) от [физических лиц](https://ru.wikipedia.org/wiki/%D0%A4%D0%B8%D0%B7%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%BB%D0%B8%D1%86%D0%BE) в режиме [самообслуживания](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%BC%D0%BE%D0%BE%D0%B1%D1%81%D0%BB%D1%83%D0%B6%D0%B8%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5). Для платежного терминала характерна высокая степень автономности его работы. Контроль за работой можно производить через [Интернет](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82). Платежные терминалы служат для приема платежей за услуги мобильной связи, коммунальные услуги, Интернет-провайдеров, в счет погашения банковских кредитов, а также для пополнения лицевых счетов в платежных системах, счетов банковских карт.

### Информационные киоски

Информационный киоск (информационный терминал, инфокиоск, инфомат) – автоматизированный программно-аппаратный комплекс, предназначенный для предоставления справочной информации. В отличие от обычного справочного киоска, электронный информационный киоск работает автономно.

Информационные киоски собирают на базе персонального компьютера, оснащенного [сенсорным монитором](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BD%D1%81%D0%BE%D1%80%D0%BD%D1%8B%D0%B9_%D0%BC%D0%BE%D0%BD%D0%B8%D1%82%D0%BE%D1%80) и установленного в [эргономичный](https://ru.wikipedia.org/wiki/%D0%AD%D1%80%D0%B3%D0%BE%D0%BD%D0%BE%D0%BC%D0%B8%D0%BA%D0%B0) вандалостойкий, как правило, стальной корпус. Помимо стали, корпуса киосков изготавливаются с применением антивандального закаленного стекла (сталинита), искусственного камня, пластика.

Дополнительно на инфокиоск может устанавливаться купюроприемник, разъемы [USB](https://ru.wikipedia.org/wiki/USB), фискальный регистратор, аудиосистема, термопринтер, дополнительный рекламный монитор, сканер штрих кодов, RFID приемник, NFC и прочее оборудование.

## Средства разработки

### WPF

Windows Presentation Foundation (WPF) – система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML.

В основе WPF лежит векторная система визуализации, не зависящая от разрешения устройства вывода и созданная с учетом возможностей современного графического оборудования. WPF предоставляет средства для создания визуального интерфейса, включая язык XAML (Extensible Application Markup Language), элементы управления, привязку данных, макеты, двухмерную и трехмерную графику, анимацию, стили, шаблоны, документы, текст, мультимедиа и оформление.

Графической технологией, лежащей в основе WPF, является DirectX, в отличие от Windows Forms, где используется GDI/GDI+. Производительность WPF выше, чем у GDI+ за счет использования аппаратного ускорения графики через DirectX.

Также существует урезанная версия CLR, называющаяся WPF/E, она же известна как Silverlight.

XAML представляет собой язык декларативного описания интерфейса, основанный на XML. Также реализована модель разделения кода и дизайна, позволяющая кооперироваться программисту и дизайнеру. Кроме того, есть встроенная поддержка стилей элементов, а сами элементы легко разделить на элементы управления второго уровня, которые, в свою очередь, разделяются до уровня векторных фигур и свойств/действий. Это позволяет легко задать стиль для любого элемента.

Для работы с WPF требуется любой .NET-совместимый язык. В этот список входит множество языков: C#, VB, C++, Ruby, Python, Delphi (Prism), Lua и многие другие. Для полноценной работы может быть использована как Visual Studio, так и Expression Blend. Первая ориентирована на программирование, а вторая — на дизайн и позволяет делать многие вещи, не прибегая к ручному редактированию XAML. Примеры этому – анимация, стилизация, состояния, создание элементов управления и так далее.

### WCF

Windows Communication Foundation (WCF) представляет собой платформу для построения сервисноориентированных приложений. С помощью WCF можно отправлять данные в виде асинхронных сообщений от одной конечной точки службы к другой.Конечная точка службы может входить в постоянно доступную службу, размещаемую в IIS, или представлять службу, размещаемую в приложении.Конечная точка может быть клиентом службы, которая запрашивает данные от конечной точки службы.Сообщения могут представлять одиночный символ или одно слово, отправляемое в формате XML, или иметь вид сложного потока двоичных данных.

Основные возможности WCF:

1. **Сервис-ориентированность.**

Применение стандартов WS в WCF позволяет создавать сервисноориентированныеприложения. Сервисноориентированная архитектура (SOA) подразумевает применение веб-служб для отправки и получения данных. Общим преимуществом служб является слабая связанность вместо жесткой запрограммированности для различных приложений. Слабая связь означает, что любой клиент, созданный на любой платформе, может подключаться к любой службе при условии, что выполняются необходимые контракты.

1. **Несколько шаблонов сообщений.**

Обмен сообщениями выполняется по одному из нескольких шаблонов. Чаще всего используется шаблон «запрос-ответ», когда одна конечная точка запрашивает данные от другой конечной точки. Вторая конечная точка отвечает. Существуют и другие шаблоны, например, одностороннее сообщение, когда одна конечная точка отправляет сообщение, не ожидая ответа. Более сложным является шаблон дуплексного обмена, когда две конечные точки устанавливают соединение и отправляют данные в обоих направлениях подобно программе обмена мгновенными сообщениями.

1. **Контракты данных.**

Поскольку платформа WCF построена на основе .NET Framework, в нее входят удобно используемые в коде методы передачи контрактов, которые нужно принудительно выполнять. Одним из универсальных типов контрактов является контракт данных. Если код службы создается на языке Visual C# или Visual Basic, то самым простым способом обработки данных фактически является создание классов, которые представляют сущность данных со свойствами, принадлежащими сущности данных. WCF включает сложную систему для работы с данными этим удобным способом. После создания классов, представляющих данные, служба автоматически создает метаданные, которые позволяют клиентам обеспечивать соответствие заданным типам данных.

1. **Безопасность.**

Сообщения можно шифровать для защиты конфиденциальности и требовать от пользователей проходить проверку подлинности перед приемом сообщений. Можно реализовать широко известные стандарты безопасности, такие как SSL и WS-SecureConversation.

1. **Несколько транспортов и кодировок.**

Сообщения могут отправляться по любому из нескольких встроенных транспортных протоколов в различных кодировках. Самым распространенным вариантом является передача сообщений SOAP в текстовой кодировке по протоколу HTTP для использования в Интернете. Кроме того, WCF позволяет отправлять сообщения по протоколу TCP, через именованные каналы или MSMQ. Сообщения можно кодировать в виде текста или использовать оптимизированный двоичный формат. Двоичные данные можно эффективно отправлять с использованием стандарта MTOM. Если ни один из предоставляемых транспортов и кодировок не подходит к текущим требованиям, можно создать собственный пользовательский транспорт или кодировку.

1. **Надежные сообщения и сообщения в очереди.**

WCF поддерживает надежный обмен сообщениями с использованием надежных сеансов, реализованных на базе схемы WS-Reliable, и обмен с использованием MSMQ.

1. **Устойчивые сообщения.**

Устойчивые сообщения не теряются в случае перебоев связи. Сообщения, передаваемые по устойчивому шаблону, всегда сохраняются в базе данных. Если происходит перебой связи, база данных позволяет возобновить обмен сообщениями после восстановления соединения. Устойчивое сообщение также можно создать с помощью Windows Workflow Foundation (WF).

1. **Поддержка AJAX и REST.**

REST — это пример развития технологии Web 2.0. WCF можно настроить для обработки "обычных" XML-данных, не упакованных в конверт протокола SOAP. WCF также можно расширить для поддержки определенных форматов XML, таких как ATOM (распространенный стандарт RSS), и даже форматов, отличных от XML, таких как нотация объектов JavaScript (JSON).

1. **Расширение среды.**

Архитектура WCF предусматривает ряд точек для расширения. Если требуются дополнительные возможности, поддерживаются точки входа, посредством которых можно настроить поведение службы.

### ASP.NET MVC

ASP.NET MVC представляет собой платформу для создания сайтов и веб-приложений с использованием паттерна (или шаблона) MVC (model – view – controller).

Работа над новой платформой была начата в 2007 году, а в 2009 году появилась первая версия. В итоге к текущему моменту уже было выпущено 5 версиq платформы, а сам фреймворк обрел большую популярность по всему миру благодаря своей гибкости и адаптивности.

Шаблон MVC, лежащий в основе новой платформы, подразумевает взаимодействие трех компонентов: контроллера (controller), модели (model) и представления (view).

Контроллер (controller) представляет класс, с которого начинается работа приложения. Этот класс обеспечивает связь между моделью и представлением. Получая вводимые пользователем данные, контроллер исходя из внутренней логики при необходимости обращается к модели и генерирует соответствующее представление.

Представление (view) - это собственно визуальная часть или пользовательский интерфейс приложения - например, html-страница, через которую пользователь, зашедший на сайт, взаимодействует с веб-приложением.

Модель (model) представляет набор классов, описывающих логику используемых данных.

Общая схема взаимодействия MVC компонентов представлена на рисунке 2.1.

****

Рисунок 2.1 – Схема взаимодействия MVC компонентов

В этой схеме модель является независимым компонентом: любые изменения контроллера или представления не затрагивают модель. Контроллер и представление являются относительно независимыми компонентами, и нередко их можно изменять независимо друг от друга.

Благодаря этому реализуется концепция разделение ответственности, в связи с чем легче построить работу над отдельными компонентами. Кроме того, вследствие этого приложение обладает лучшей тестируемостью.

### Microsoft SQL Server

Microsoft SQL Server — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

Преимущества MS SQL Server:

– обладает высокой степенью производительности и отказоустойчивости;

– является многопользовательской СУБД и работает по принципу «клиент-сервер»;

– клиентская часть системы поддерживает создание пользовательских запросов и их отправку для обработки на сервер.

– тесная интеграция с операционной системой Windows;

– поддержка удаленных подключений;

– поддержка популярных типов данных, а также возможность создания триггеров и хранимых процедур;

– встроенная поддержка ролей пользователей;

– расширенная функция резервного копирования баз данных;

– высокая степень защищенности.

# ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К СИСТЕМЕ

## Требования к функциональным характеристикам системы

Система должна разрабатываться как готовый продукт. Функциональность системы должна обеспечиваться организацией взаимосвязей следующих программных средств:

– ПС терминала;

– ПС сервера;

– ПС бизнес администратора.

ПС терминала должно включать в себя:

– программные интерфейсы обмена данными между функциональными компонентами ПС терминала;

– программные интерфейсы обмена данными с программным средством сервера;

– графические интерфейсы и функционал для взаимодействия с пользователем ВОТ.

ПС терминала должно предоставлять интерфейсы взаимодействия для следующих пользователей ВОТ:

– лиц, совершающих обменную операцию (далее клиент);

– лиц, осуществляющих сервисное обслуживание ВОТ и инкассацию.

ПС сервера должно включать в себя:

– программные интерфейсы и функционал для обмена данными с программным средством терминала;

– программные интерфейсы и функционал обмена данными с программным средством бизнес администратора;

ПС бизнес администратора должно включать в себя:

– программные интерфейсы обмена данными с программным средством сервера;

– графические интерфейсы взаимодействия с пользователями ПС бизнес администратора (управление терминалами, пользователями и курсами валют, просмотр статистической и формирование отчетной информации).

## Требования к составу и параметрам технических средств ВОТ

Для работы ПС терминала ВОТ должен быть оборудован и укомплектован следующими устройствами:

– купюроприемник модель iVIZION-100-HC, со встроенным микропрограммным обеспечением, позволяющим принимать три валюты: рубли, доллары и евро;

– диспенсер модели HMD;

– система видеонаблюдения;

– чековый принтер модели VKP 80 II;

– датчики модели APC 750;

– источник бесперебойного питания;

– монитор с сенсорным экраном и диагональю 17 дюймов.

Система видеонаблюдения должна быть оснащена тремя ip-видеокамерами ZAVIO F3101:

– фронтальной камерой;

– камерой купюроприемника;

– камерой диспенсенра.

В ВОТ должны быть установлены следующие датчики:

– датчик открытия двери ВОТ;

– температурный датчик;

– датчик удара;

– датчик открытия сейфа;

– датчик наклона.

Для обеспечения работы ПС терминала необходимо обеспечить следующие минимальные требования к блоку системной памяти ВОТ:

– 64-разрядный процессор Intel с тактовой частотой 3,4 ГГц;

– объем свободной оперативной не менее памяти 8 Гб;

– сетевая карта;

– операционная система Windows 7/8/8.1.

## Требования к ПС терминала

Данное программное средство должно обеспечивать:

– проведение операций обмена валют по курсу, установленному ПС бизнес администратора для ВОТ;

– возможность сервисного обслуживания и инкассации ВОТ;

– отображение текущего режима работы ВОТ;

– отображение информации о курсах обмена валют в ВОТ;

– отображение информации о купюрах, принимаемых ВОТ к обмену;

– отображение информации о состоянии оборудования, входящего в состав ВОТ;

– отображение информации по счетчикам кассет диспенсера и купюроприемника;

– отображение рекламной информации.

ПС терминала должно включать в себя функционал для входа в режим сервисного обслуживания и инкассацию. Данный функционал должен обеспечивать:

– интерфейсы для ввода пользователем логина и пароля;

– формирование запроса к ПС сервера для аутентификации пользователя;

– переход в режим сервисного обслуживания и инкассации при успешной аутентификации пользователя ПС сервера.

ПС терминала должно включать в себя:

– программные компоненты, обеспечивающие взаимодействие с аппаратно-программными устройствами (далее оборудованием), входящими в состав ВОТ;

– управляющую программу;

Управляющая программа должна обеспечивать:

– взаимодействие между программными компонентами ПС терминала;

– взаимодействие с ПС сервера;

– взаимодействие с пользователем ВОТ.

ПС терминала должно включать программные компоненты для взаимодействия с оборудованием, входящим в состав ВОТ:

– программный компонент купюроприемника;

– программный компонент диспенсера;

– программный компонент системы видеонаблюдения;

– программный компонент чекового принтера;

– программный компонент блока датчиков;

– программный компонент источника бесперебойного питания.

Вышеперечисленные программные компоненты должны предоставлять программные интерфейсы для взаимодействия с ними.

## Требования к надежности

При возникновении внештатных ситуаций система должна выполнять следующие действия для сохранения работоспособности и восстановления своих функций:

– при сбоях в системе электроснабжения аппаратной части корректно завершать действие, выполняемое ПС терминала и перевод ВОТ в режим не обслуживания.

– при сбоях в системе электроснабжения аппаратной части, приводящих к перезагрузке ОС, ПС терминала должно пытаться восстановить работу ВОТ по обслуживанию пользователей после перезапуска ОС;

– при ошибках и сбоях в работе аппаратных средств пытаться восстановить их работоспособность;

– при неудачном восстановлении работоспособности аппаратных средств переводить ВОТ в режим не обслуживания.

Для защиты аппаратных средств системы от скачков и перепадов напряжения в состав технических средств входит ИБП, который обеспечивает переключение на питание батареи ИБП для корректного завершения действия, выполняемого системой.

При ошибках, связанных с системным программным обеспечением, восстановление работоспособности возлагается на ОС.

Уведомлять ПС сервера обо всех внештатных ситуациях, произошедших на ВОТ.

# ПРОЕКТИРОВАНИЕ СИСТЕМЫ

## Выбор методологий моделирования и инструментария

Для визуального моделирования проблемной области было отдано предпочтение Rasional Rose компании Rational Software. Данное средство является простым и полностью интегрированным решением для разработки ПО, включая Интернет-решения. Rational Rose является стандартом дефакто среди инструментов проектирования приложений. Ни одно другое CASE-средство не предлагает такую широту и глубину решений как платформа Rational. С помощью Rational Rose можно визуализировать, изменять и тестировать модель.

Одно из неоспоримых преимуществ Rational Rose – обратное проектирование, поскольку разработчику и проектировщику важно увидеть перед изменениями уже работающую систему в нормальном графическом представлении. Как правило визуально-графический ряд оказывает куда большее воздействие нежели пролистывание технических заданий и программных текстов. Тем более что, проект, подвергшийся обратному проектированию может быть доработан и вновь сгенерирован (а впоследствии и скомпилирован). Rational Rose предоставляет для этого все необходимые средства.

Rational Rose является лидирующим инструментом визуального моделирования, поскольку он имеет все необходимые возможности: поддержку UML, многоязыковую поддержку итерационной разработки, полную поддержку командной разработки, компонентно-базированную разработку с поддержкой ведущих архитектур и таких компонентных моделей, как WinDNA и J2EE/SE/ME, легкость применения, оптимизированную интеграцию и многое другое.

Для проектирования и моделирования данных был использован инструментарий AllFusion ERwin Data Modeler (ERwin) компании Computer Associates. ERwin позволяет проектировать, документировать и сопровождать базы данных, хранилища данных и витрины данных (data marts). Основные аргументы и факты для разработчиков ПО в пользу использования данного инструментария:

– поддерживается прямое (создание БД на основе модели) и обратное (генерация модели по имеющейся базе данных) проектирование для 20 типов СУБД;

– увеличивает производительность труда благодаря удобному интерфейсу и автоматизации рутинных процедур;

– ERwin является стандартом де-факто;

– позволяет повторно использовать компоненты созданных ранее моделей, а также использовать наработки других разработчиков. Повышается эффективность;

– позволяет переносить структуру БД из СУБД одного типа в СУБД другой;

– позволяет документировать структуру БД;

– продукт можно использовать на всех стадиях жизненного цикла баз данных;

– позволяет получить точную и наглядную информацию, где хранятся данные и как получить к ним доступ;

– позволяет, используя визуальные средства, описать структуру БД, а затем автоматически сгенерировать файлы данных для любого типа СУБД.

## Разработка диаграмм вариантов использования

Диаграмма вариантов использования описывает функциональное назначение системы, т.е. то, что система будет делать в процессе своего функционирования, и является исходной концептуальной моделью системы в процессе ее проектирования и разработки.

Разработка диаграммы вариантов использования преследует следующие цели:

– определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы, а также сформулировать общие требования к функциональному поведению проектируемой системы;

– разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей;

– подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью так называемых вариантов использования. При этом актером (actor) называется любая сущность, взаимодействующая с системой извне. В свою очередь, вариант использования (use case) служит для описания сервисов, которые система предоставляет актеру, т.е. каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актером. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие актеров с системой.

### Действующие лица

Множество систем имеет разделение на категории пользователей. Поэтому каждая категория пользователей представляется отдельным действующим лицо (актером).

При анализе работы системы были выделены следующие действующие лица и разработаны варианты использования:

Клиент – может совершать валютно-обменные операции.

Инкассатор – может производить инкассацию.

Бизнес администратор – может управлять терминалами, курсами валют, по которым производится обмен, а также купюрами, доступными к приему.

### Варианты использования

Были выделены следующие варианты использования:

– обмен валюты;

– смена языка интерфейса;

– вход в сервисный режим;

– печать счетчиков;

– проведение инкассации;

– аутентификация в АРМ;

– смена курсов валют.

### ­Диаграмма вариантов использования

Диаграммы вариантов использования являются необходимым средством при анализе требований, планировании и управлении итеративной разработкой. Работа с вариантами использования является одной из самых важных на стадии уточнения. Каждый вариант использования – это потенциальное требование к системе, и пока оно не выявлено, невозможно запланировать его реализацию.

При анализе задач и требований, поставленных при разработке программного комплекса, была спроектирована диаграмма вариантов использования для ПС терминала, которая приводится на рисунке 4.1.

На данной диаграмме отображены основные варианты использования, необходимые разрабатываемому ПС. Так, инкассатор может производить инкассацию. Любой пользователь может совершить обменную операцию.

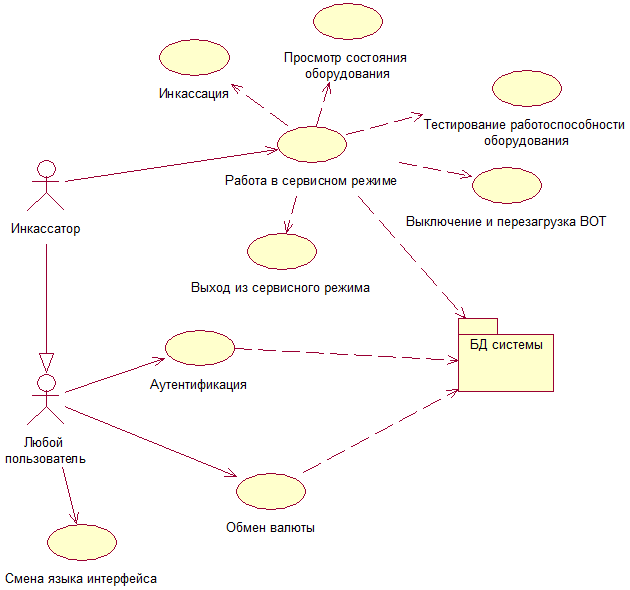


Рисунок 4.1 – Диаграмма вариантов использования ПС терминала

### Описание вариантов использования

*Вариант использования «Обмен валюты»:*

– назначение: служит для проведения транзакции по обмену валют;

– основное действующее лицо: клиент;

– предусловия: нет;

– основной сценарий:

1) клиент вставляет купюру в купюроприемник;

2) если купюра распознана, то

2.1) система производит расчет, исходя из курса обмена;

2.2) клиенту выводится окно с расчетом по транзакции;

2.3) если клиент подтвердил транзакцию, то

2.3.1) производится набор банкнот к выдаче;

2.3.2) если набор успешен, то

2.3.2.1) окончательное «проглатывание» купюры;

2.3.2.2) выдача денег;

2.3.2.3) печать чека;

2.3.3) иначе:

2.3.3.1) выдается сообщение о невозможности операции;

2.3.3.2) возвращается купюра;

2.4) иначе, возвращается купюра;

3) иначе:

3.1) выдается сообщение о невозможности операции;

3.2) возвращается купюра;

– постусловия: система регистрирует завершение транзакции на сервере.

*Вариант использования «Вход в сервисный режим»:*

– назначение: служит для входа в сервисный режим для проведения инкассации либо просмотра состояния оборудования;

– основное действующее лицо: инкассатор;

– предусловия: инкассатор нажал на секретную кнопку для отображения авторизационной формы;

– основной сценарий:

1) система запрашивает логин и пароль;

2) клиент вводит логин и пароль;

3) если авторизационные данные верны, то происходит вход в сервисный режим;

4) иначе:

4.1) если количество вводов неверного пароля меньше 4, то

4.1.1) выдать предупреждение о неправильном пароле;

4.1.2) вернуться к началу сценария;

4.2) иначе:

4.2.1) выдать предупреждение о неправильном пароле;

4.2.2) уведомить сервер о количестве неверных вводов пароля;

4.2.3) вернуться к началу сценария;

– постусловия: нет.

*Вариант использования «Печать счетчиков»:*

– назначение: служит для получения информации о состояниии кассет диспенсера и купюроприемника;

– основное действующее лицо: инкассатор;

– предусловия: перед началом выполнения данного варианта использования «Инкассатор» должен войти в сервисный режим;

– основной сценарий:

1) инкассатор нажимает на кнопку «Печать счетчиков» на вкладке «Инкассация» сервисного режима;

2) если принтер находится в рабочем состоянии, то

2.1) если в чековом слоте находится чек, то

2.1.1) попытаться втянуть чек;

2.2) попытаться распечатать счетчики;

2.3) если печать успешна, то

2.3.1) разблокировать кнопку «Начать инкассацию».

2.4) иначе, выдать сообщение об ошибке;

3) иначе, выдать сообщение о неработоспособности принтера.

*Вариант использования «Проведение инкассации»:*

*–* назначение: служит для проведения инкассации;

– основное действующее лицо: инкассатор;

– предусловия: перед началом выполнения данного варианта использования необходимо выполнить «Печать счетчиков»;

– основной сценарий:

1) система предлагает начать инкассацию;

2) инкассатор нажимает на кнопку «Начать инкассацию»;

3) кассеты диспенсера разблокируются;

4) если разблокировка успешна, то:

4.1) инкассатор производит замену кассет;

4.2) инкассатор вводит данные о количестве купюр для каждой кассеты;

4.3) инкассатор нажимает на кнопку «Завершить инкассацию»;

4.4) если все данные о кассетах заполнены, то:

4.4.1) выдается сообщение об успешной инкассации;

4.5) иначе, выдается сообщение «Заполните все данные по кассетам»;

5) иначе, выдать сообщение «Не удалось разблокировать кассеты диспенсера»;

– постусловия: нет.

*Вариант использования: «Аутентификация в АРМ»:*

– назначение: служит для аутентификации и доступа бизнес администратора к АРМ;

– основное действующее лицо: бизнес администратор;

– предусловия: запрошена страница аутентификации либо недоступная для аутентифицированного пользователя страница;

– основной сценарий:

1) система запрашивает логин и пароль;

2) бизнес администратор вводит логин и пароль;

3) если авторизационные данные верны, то аутентификация успешна;

4) иначе:

4.1) выдать предупреждение о неправильном пароле;

4.2) вернуться к началу сценария;

– постусловия: нет.

*Вариант использования: «Смена курсов валют»:*

– назначение: служит для изменения курсов валют;

– основное действующее лицо: бизнес администратор;

– предусловия: нет;

– основной сценарий:

1) переход к подчиненному сценарию «Аутентификация на ВОТ»;

2) если аутентификация прошла успешно, то

2.1) переход на страницу смены курсов;

2.2) бизнес администратор выбирает валюту и устанавливаемый курс для выбранной валюты;

2.3) бизнес администратор выбирает дату и время, с которого будет действовать установленный курс [опционально];

2.4) бизнес администратор нажимает на кнопку «Установить курс»;

2.5) если дата установлена и меньше текущей, выдать сообщение об ошибке;

2.6) иначе, курс успешно установлен.

– постусловия: нет.

## Построение логической модели данных

Перед проектированием физической модели данных необходимо смоделировать логическую модель. Логическая модель описывает всю базу данных как единое целое. Кроме того, необходимое пользователю логическое представление данных может существенно отличаться от общей модели данных.

Проанализировав работу пользователя с программным комплексом, проанализировав входные и выходные данные, была построена логическая модель данных на уровне сущностей, которая представлена на рисунке 4.2.

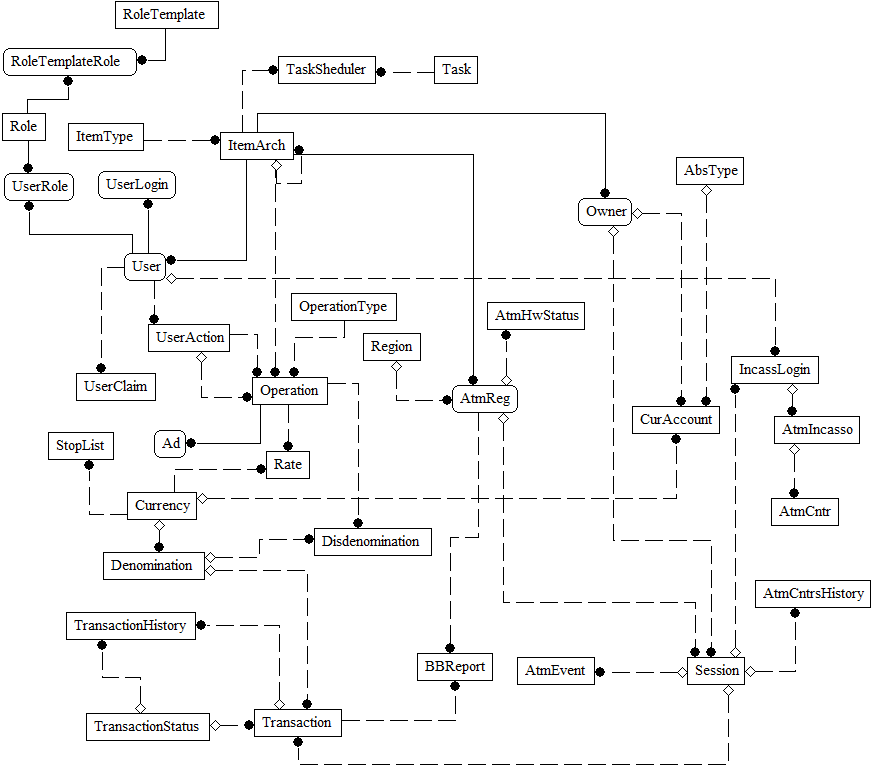


Рисунок 4.2 – Логическая модель данных на уровне сущностей

# ФИЗИЧЕСКОЕ МОДЕЛИРОВАНИЕ

## Физическая модель данных

Физическая модель данных описывает реализацию объектов логической модели на уровне объектов конкретной базы данных.

### Проектирование и создание таблиц базы данных

На основании логической модели базы данных на рисунке 5.1 приводится физическая структура таблиц и их связей базы данных процессора правил, реализованная с помощью СУБД Microsoft SQL Server 2014.

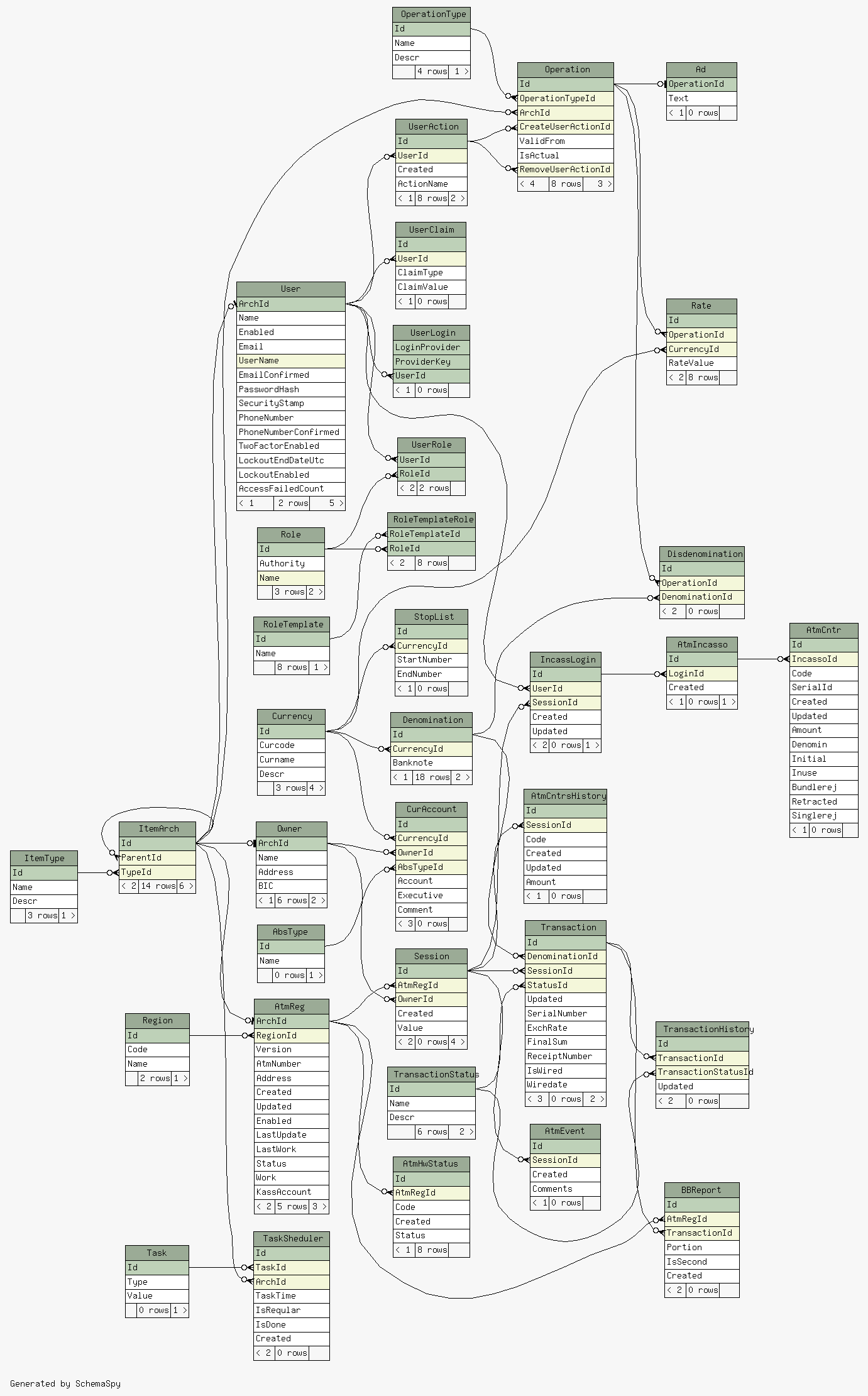


Рисунок 5.1 – Физическая модель данных

## Компоненты предмета разработки

Диаграмма компонентов отражает общие зависимости между компонентами.

Описанные ниже диаграммы компонентов разработаны для следующих целей:

– визуализации общей структуры исходного кода программной системы;

– обеспечения многократного использования отдельных фрагментов кода.

Разрабатываемая система на уровне пользователя состоит из трех частей: клиента, инкассатора и бизнес администратора, а на физическом уровне – из базы данных (размещается на сервере информационного ресурса и является одним из компонентов СУБД), ПС бизнес администратора в виде отображаемых в браузере страниц и ПС терминала в виде настольного приложения.

Диаграмма компонентов ПС терминала приводится на рисунке 5.2.

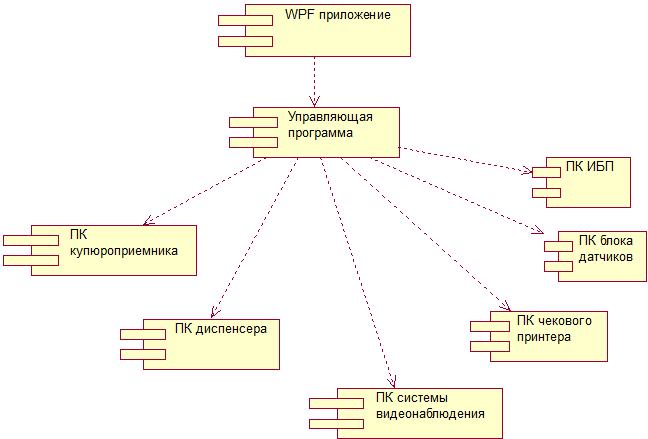


Рисунок 5.2 – Диаграмма компонентов ПС терминала

Диаграмма компонентов ПС сервера изображена на рисунке 5.3.

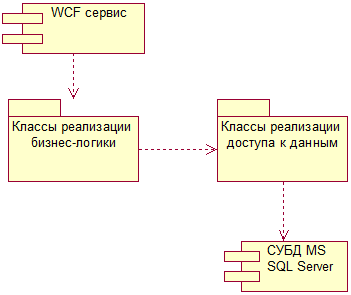


Рисунок 5.3 – Диаграмма компонентов ПС сервера

Диаграмма компонентов ПС бизнес администратора приводится на рисунке 5.4.

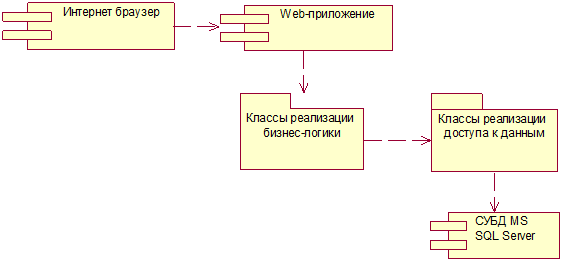


Рисунок 5.4 – Диаграмма компонентов ПС бизнес администратора

## Описание компонентов ПС терминала

### ПК купюроприемника

Программный компонент купюроприемника взаимодействовует:

– с устройством купюроприемника по интерфейсам, предоставленным поставщиком купюроприемника;

– с управляющей программой ВОТ по интерфейсам, разработанным в рамках данного проекта.

Функции программного компонента купюроприемника:

– осуществлять обработку сообщений от устройства купюроприемника;

– уведомлять управляющую программу о событиях и сбоях купюроприемника;

– формировать команды купюроприемнику в соответствии с командами, полученными от управляющей программы ВОТ;

– возвращать управляющей программе результаты выполнения купюроприемником ее команд.

Программный компонент купюроприемника уведомляет управляющую программу ВОТ о следующих событиях устройства купюроприемника:

– купюра не распознана и возвращена (нераспознанная купюра возвращена пользователю);

– распознанная купюра возвращена (пользователь отказался от обмена, купюра возвращена пользователю);

– купюроприемник в состоянии ожидания купюр.

Программный компонент купюроприемника обеспечивает выполнение устройством купюроприемника следующих команд управляющей программы ВОТ:

– выполнить первоначальную инициализацию устройства купюроприемника;

– выполнить сброс устройства;

– перевести устройство в режим ожидания купюр;

– перевести устройство в режим запрета приема купюр;

– сбросить купюру из устройства в сбросовую кассету;

– вернуть купюру из устройства пользователю;

– удерживать купюру в устройстве (не возвращать и не сбрасывать в кассету);

– предоставить статус устройства;

– предоставить отсканированное устройством изображение купюры;

– запретить к приему устройством указанные номиналы купюр;

– предоставить код валюты и номинал распознанной купюры.

### ПК диспенсера

Программный компонент диспенсера обеспечивает взаимодействие управляющей программы ВОТ с устройством диспенсера ВОТ для набора купюр из кассет диспенсера (белорусские рубли) и выдачу набранных купюр клиенту. При получении команды от управляющей программы о прекращении операции обмена программный компонент диспенсера выполняет сброс набранных диспенсером купюр в сбросовую кассету ВОТ.

Программный компонент диспенсера взаимодействовует:

– с устройством диспенсера по интерфейсам, предоставленным поставщиком диспенсера;

– с управляющей программой ВОТ по интерфейсам, разработанным в рамках данного проекта.

Функции программного компонента диспенсера:

– уведомлять управляющую программу о событиях и сбоях диспенсера;

– формировать команды диспенсеру в соответствии с командами, полученными от управляющей программы ВОТ;

– возвращать управляющей программе результаты выполнения диспенсером ее команд.

Программный компонент диспенсера обеспечивает выполнение устройством диспенсера следующих команд управляющей программы ВОТ:

– вернуть состояния счетчиков;

– вернуть идентификатор кассеты;

– набрать заданное количество купюр из указанной кассеты;

– сбросить набранные купюры в сбросовую кассету;

– выдать набранные купюры клиенту;

– обнулить счетчики (режим сервисного обслуживания и инкассации ВОТ);

– выполнить открытие/закрытие кассет (режим сервисного обслуживания и инкассации ВОТ).

### ПК системы видеонаблюдения

Программный компонент системы видеонаблюдения обеспечивает:

– взаимодействие с управляющей программой ВОТ по интерфейсам, разработанным в рамках данного проекта;

– взаимодействие с камерами для получения фрагментов записи по интерфейсам, предоставленным поставщиком камер видеонаблюдения;

– сохранение данных с цифровых камер в графическом файле.

Программный компонент системы видеонаблюдения взаимодействовует с управляющей программой ВОТ по интерфейсам обмена данными между программными компонентами ВОТ, разработанными в рамках данного проекта.

Функции программного компонента системы видеонаблюдения:

– уведомлять управляющую программу о событиях и сбоях камер видеонаблюдения;

– формировать команды камерам видеонаблюдения в соответствии с командами, полученными от управляющей программы ВОТ;

– возвращать управляющей программе статус выполнения камерами видеонаблюдения ее команд.

Программный компонент системы видеонаблюдения обрабатывает и выполняет следующие команды от управляющей программы ВОТ:

– инициализировать цифровые камеры, установленные в ВОТ;

– начать фиксацию данных с указанной камеры;

– завершить фиксацию данных с указанной камеры.

Данные, зафиксированные программным компонентом системы видеонаблюдения сохраняются в графических файлах с заданной периодичностью. Имя файла содержит номер цифровой камеры, дату и время, включая секунды. Интервал фиксации данных включает периоды временит до наступления события и после окончания события. После успешного выполнения операции сохранения фрагментов данных в графических файлах программный компонент системы видеонаблюдения передает управляющей программы ВОТ путь к сохраненным графических файлам.

### ПК чекового принтера

Программный компонент чекового принтера обеспечивает взаимодействие управляющей программы ВОТ с устройством чекового принтера ВОТ для распечатки на бумажном носителе:

– чека по каждой успешно проведенной ВОТ операции обмена валюты;

– значения счетчиков диспенсера и купюроприемника в режиме сервисного обслуживания и инкассации ВОТ.

Программный компонент чекового принтера взаимодействовует:

– с устройством чекового принтера по интерфейсам, предоставленным поставщиком данного устройства;

– с управляющей программой ВОТ по интерфейсам, разработанным в рамках данного проекта.

Программный компонент чекового принтера обеспечивает выполнение устройством чекового принтера следующих команд управляющей программы ВОТ:

– печать чека по валютно-обменной операции;

– печать значения счетчиков устройств диспенсера и купюроприемника;

– сообщить статус состояния устройства чекового принтера.

При получении команды "Печать чека валютно-обменной операции" программный компонент чекового принтера проверяет готовность к печати устройства чекового принтера. Для проверки готовности к печати программный компонент взаимодействовует с чековым принтером, формирует запросы к устройству и обрабатывает ответы от чекового принтера. Данный программный компонент формирует чековому принтеру следующие команды:

– проверить наличие бумаги в устройстве чекового принтера;

– проверить наличие в устройстве чекового принтера чека, оставленного пользователем;

– удалить из устройства чекового принтера чек, оставленный пользователем.

В случае отсутствия бумаги в устройстве чекового принтера программный компонент чекового принтера уведомляет управляющую программу ВОТ об этом событии.

Программный компонент чекового принтера выполняет:

– возврат управляющей программе ВОТ результатов выполнения команд;

– сообщение управляющей программе ВОТ о сбоях чекового принтера и ошибках, произошедших во время выполнения команд.

### ПК блока датчиков

Данный программный компонент осуществляет мониторинг всех датчиков, входящих в состав ВОТ. Программный компонент блока датчиков снимает показания с датчиков, установленных в ВОТ и уведомляет управляющую программу ВОТ о критических изменениях в показаниях датчиков.

Программный компонент блока датчиков обеспечивает:

– взаимодействие с управляющей программой ВОТ по интерфейсам, разработанным в рамках данного проекта;

– взаимодействие с блоком датчиков по интерфейсам, предоставленным поставщиком датчиков.

Функции данного программного компонента:

– осуществлять мониторинг всех датчиков ВОТ;

– снимать показания и контролировать изменения состояния датчиков (датчик температуры, наклона);

– получать уведомления о срабатывании датчиков ВОТ (датчик удара, датчик открытия сейфа, датчик открытия дверей).

При отклонении состояния датчика от критического значения (датчик температуры, наклона) или при получении сигнала о срабатывании датчиков (датчик удара, датчик открытия сейфа, датчик открытия дверей), программный компонент блока датчиков уведомляет об этих событиях управляющую программу ВОТ.

### ПК источника бесперебойного питания

Данный программный компонент обеспечивает мониторинг источника бесперебойного питания, входящего в состав ВОТ. Программный компонент источника бесперебойного питания обеспечивает взаимодействие управляющей программы ВОТ с устройством источника бесперебойного питания (далее ИБП) ВОТ и уведомляет управляющую программу ВОТ о событиях ИБП. Программный компонент ИБП обеспечивает:

– взаимодействие с управляющей программой ВОТ по интерфейсам, разработанным в рамках данного проекта;

– взаимодействие с ИБП по интерфейсам, предоставленным поставщиком данного устройства.

Функции данного программного компонента:

– осуществлять мониторинг ИБП ВОТ;

– оповещать управляющую программу ВОТ о состоянии и текущем режиме работы ИБП.

По запросу управляющей программы программный компонент источника бесперебойного питания возвращает время работоспособности ВОТ с использованием батареи ИБП при текущей нагрузке. Программный компонент должен уведомляет управляющую программу о подключении/отключении ИБП.

### Управляющая программа ВОТ

Управляющая программа ВОТ обеспечивает:

– взаимодействие и управление программными компонентами ПС терминала;

– взаимодействие с ПС сервера;

– взаимодействие с пользователем ВОТ;

– ведение журналов работы для всех программных компонентов валютно-обменного терминала.

Управляющая программа ВОТ включает в себя:

– программные интерфейсы обмена данными между программными компонентами ПС терминала;

– программные интерфейсы обмена данными с ПС сервера;

– интерфейсы взаимодействия с пользователем.

Интерфейсы взаимодействия с пользователем включают:

– интерфейсы взаимодействия с клиентом;

– интерфейсы взаимодействия с лицом, осуществляющим сервисное обслуживание ВОТ и инкассацию.

Управляющая программа ВОТ обеспечивает функционирование ВОТ в следующих режимах:

– режим загрузки;

– режим обслуживания;

– режим сервисного обслуживания и инкассации ВОТ;

– режим не обслуживания.

Режим обслуживания включает в себя следующие подрежимы:

– подрежим ожидания пользователя;

– подрежим валютно-обменной операции.

В режиме сервисного обслуживания и инкассации ВОТ управляющая программа:

– предоставляет пользователю возможность выполнить авторизацию для перехода валютно-обменного терминала в режим сервисного обслуживания и инкассации;

– информирует пользователя о невозможности переключения ВОТ в режим сервисного обслуживания и инкассации (пользователь не авторизирован в ССВОО);

– переключает ВОТ в режим сервисного обслуживания и инкассации ВОТ (пользователь авторизирован в ССВОО).

## Развертывание предмета разработки

Для визуализации элементов и компонентов системы, существующих лишь на этапе ее исполнения, на рисунке 5.5 приводится диаграмма развертывания с изображением элементов, из которых должна состоять система. На диаграмме изображены только компоненты-экземпляры программы, являющиеся исполняемыми файлами или динамическими библиотеками.

G:\kate\Графика\Untitled Diagram (1).png

Рисунок 5.5 – Диаграмма развертывания системы

В отличие от диаграмм логического представления, диаграмма развертывания является единой для системы в целом, поскольку всецело отражает особенности ее реализации. Эта диаграмма, завершает процесс объектно-ориентированного анализа и проектирования для данной программной системы, и ее разработка является последним этапом спецификации модели.

Перечислим цели, которые преследовались при разработке диаграммы:

– определить распределение компонентов системы по ее физическим узлам;

– показать физические связи между всеми узлами системы на этапе ее исполнения;

– выявить узкие места системы и реконфигурировать ее топологию для достижения требуемой производительности.

Браузер Web-клиента должен обязательно поддерживать библиотеку javascript-функций, иначе некоторые возможности, предоставляемые программным комплексом, будут недоступны при его работе.

# РЕАЛИЗАЦИЯ СИСТЕМЫ

## Назначение и описание компонентов ПО

Разработанная подсистема состоит из трех компонентов: настольного WPF приложения, WCF-сервиса и ASP.NET MVC 5 веб-приложения. WPF приложение необходимо для обслуживания клиентов, проведения инкассации, а также для работы с оборудованием, входящим в состав терминала. WCF-сервис отвечает за работу с терминалами, обработку клиентских запросов и сохранение информации. Веб-приложение служит для администрирования базы данных, в том числе просмотра и изменения курсов валют и списка принимаемых банкнот. Для реализации этой функциональности были использованы многие сервисы, предоставляемые ASP.NET и технология доступа к данным EntityFramework, которая входит в состав Microsoft .NET Framework 4.5.

На рисунке 6.1 приводится список разработанных проектов из обозревателя решений среды разработки Visual Studio.

В папке ArmNet находятся проекты реализации ПС бизнес администратора. Проект ArmUi представляет собой Asp.Net MVC приложение, использующееся для реализации пользовательского интерфейса. В проекте ArmBusinessLogic содержится бизнес логика приложения. Проект DataAccess представляет собой логику доступа к данным.

Папка ServerNet содержит проекты реализации ПС сервера. Проект TerminalServices содержит интерфейс и реализацию методов WCF-сервиса. Бизнес-логика сервиса вынесена в отдельную библиотеку Server.

В папке TerminalNet располагаются проекты реализации ПС терминала.

Подпапка Cpp содержит библиотеки для работы с устройствами терминала, написанные на языке C++. Проект IPCamLib содержит логику работы с камерами. Библиотека JcmUsbReset используется для взаимодействия с источником бесперебойного питания. Проект NMD100Protocol обеспечивает работу с диспенсером.

В подпапке Devices находятся библиотеки для управления компонентами терминала. Проект CashIn представляет собой библиотеку для работы с купюроприемником. Проект cctv содержит «обертку» над библиотекой IPCamLib для управления камерами. Проект Dispenser представляет собой «обертку» над библиотекой NMD100Protocol. В библиотеке Printer располагается логика работы с устройством принтера. Проект Sensors служит для получения и обработки уведомлений от датчиков удара, наклона, температуры, открытия дверей. Библиотека UPS представляет собой «обертку» над библиотекой «JcmUsbReset».

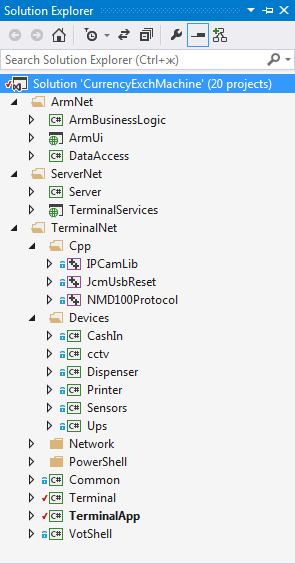


Рисунок 5.1 ­– Структура решения

Подпапка Network содержит проекты, обеспечивающие взаимодействие между приложениями посредством REST и MQ.

В подпапке Powershell находятся скрипты, использующиеся при начальной настройке ПС терминала.

Библиотека Common содержит код, общий для проектов, использующихся в ПС терминала, в том числе код для описания интерфейсов взаимодействия между компонентами.

Проект Terminal представляет собой управляющую программу, взаимодействующую с оборудованием и реализующую основную логику работы приложения.

Проект TerminalApp служит для взаимодействия с пользователями. Это приложение, разработанное с использованием технологии WPF, содержит реализацию пользовательского интерфейса посредством страниц, написанных и использованием языка разметки XAML.

Проект VotShell предсталяет собой WPF приложение, использующееся в качестве Windows Explorer для обеспечения защиты от несанкционированного доступа к операционной системе.

Для инициализации компонентов приложений используется IoC контейнер Autofac. Данный контейнер для управления зависимостями интересен тем, что кроме базового функционала, который присущ всем IoC контейнерам, существует несколько отдельных библиотек, которые позволяют управлять зависимостями для ASP.NET MVC/Web API, WCF Integration и WebApiIntegration.

На рисунке 6.2 приводится диаграмма классов данных ПС терминала.

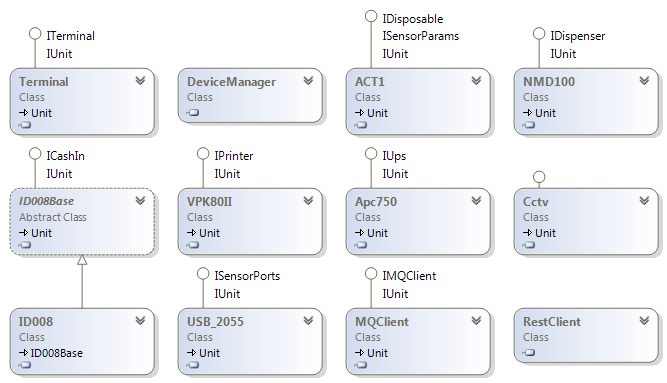


Рисунок 6.2 – Диаграмма классов данных ПС терминала

Пользовательский интерфейс ПС терминала содержит следующие файлы, написанные на языке разметки XAML:

Pages/changeRates – страница, уведомляющая пользователя о смене курсов валют;

Pages/collection – страница сервисного режима, позволяющая распечатать счетчики и начать инкассацию;

Pages/loading – страница загрузки терминала;

Pages/login – страница входа в сервисный режим;

Pages/result – страница с результатом транзакции(приглашение забрать выданные деньги либо уведомление об отмене транзакции);

Pages/transaction – страница с расчетом по транзакции;

Pages/unservicing – страница необслуживания ВОТ;

Pages/waiting – страница ожидания клиента;

Controls/BOTHeader – хедер приложения в режимах, отличных от сервисного;

Controls/BOTServiceHeader – хедер приложения в сервисном режиме;

Controls/BOTFooter – футер приложения;

Controls/CassettesInfo – область сервисного режима, содержащая информацию о состоянии кассет диспенсера;

Controls/DevicesStatus – область сервисного режима с информацией о состоянии оборудования;

Controls/InkassoStep1 – контрол разблокировки кассет диспенсера и начала инкассации;

Controls/InkassoStep2 – контрол установки счетчиков диспенсера при проведении инкассации;

Controls/InkassoStep3 – контрол, содержащий информацию об успешно завершенной инкассации;

Controls/KeyboardControl – контрол, представляющий собой виртуальную клавиатуру;

Controls/LoginControl – контрол входа в сервисный режим, содержащий поля ввода логина и пароля;

Controls/NotesControl – область режима ожидания клиента, представляющая информацию о купюрах, доступных и запрещенных к приему;

Controls/RatesControl – область режима ожидания клиента, содержащая информацию о текущих курсах валют, установленных на ВОТ.

Для реализации пользовательского интерфейса ПС бизнес администратора были созданы следующие HTML страницы на основе движка Razor:

Views\Shared\Layout.cshtml – шаблонная страница для реализации однотипного интерфейса;

Views\Account\Login.cshtml – страница входа в Web-приложение;

Views\Account\Register.cshtml – страница регистрации пользователей;

Views\Account\Index.cshtml – страница, содержащая список существующих пользователей и ссылки на дальнейшие действия с пользователями;

Views\Account\Manage.cshtml – страница редактирования пользователей;

Views\Advertisement\AdvertisementTab.cshtml – страница установки рекламы для отображения на ВОТ;

Views\Denomination\DenominationTab.cshtml – страница установки банкнот, разрешенных и запрещенных к приему;

Views\Owner\Index.cshtml – страница, содержащая список существующих владельцев терминалов и ссылки на дальнейшие действия с владельцами;

Views\Owner\Add.cshtml – страница добавления новых владельцев;

Views\Owner\Edit.cshtml – страница редактирования существующих владельцев;

Views\Rate\RateTab.cshtml – страница установки курсов валют;

Views\Region\Index.cshtml – страница, содержащая список существующих регионов и ссылки на дальнейшие действия с регионами;

Views\Region\Add.cshtml – страница добавления новых регионов;

Views\Region\Edit.cshtml – страница редактирования существующих регионов;

Views\Role\Index.cshtml – страница, содержащая список существующих ролей и ссылки на дальнейшие действия с ролями;

Views\Role\Add.cshtml – страница добавления новых ролей;

Views\Role\Edit.cshtml – страница редактирования существующих ролей;

Views\Terminal\Index.cshtml – страница, содержащая список существующих терминалов и ссылки на дальнейшие действия с ними;

Views\Terminal\Add.cshtml – страница добавления новых терминалов;

Views\Terminal\Edit.cshtml – страница редактирования существующих терминалов;

Views\Terminal\Details.cshtml – страница просмотра детальной информации о терминале;

Views\Terminal\\_FindTab.cshtml – страница поиска терминалов по параметрам;

Views\Terminal\\_Hardware.cshtml – страница просмотра информации о состоянии устройств терминала;

Views\Terminal\\_RateTab.cshtml – страница просмотра текущих и запланированных курсов валют для терминала;

Views\Terminal\\_TransactionTab.cshtml – страница просмотра информации о транзакциях, произведенных на терминале;

Views\Transaction\Index.cshtml – страница просмотра информации о произведенных транзакциях в сети терминалов;

Данные приложения имеют модульную структуру с возможностью дальнейшего расширения (дополнения новыми страницами-модулями) и модификации. Следовательно, данная система в дальнейшем легка как для сопровождения, так и для расширения.

## Режимы работы ВОТ

ПС терминала обеспечивает следующие режимы работы ВОТ:

– режим загрузки;

– режим обслуживания;

– режим сервисного обслуживания и инкассации ВОТ;

– режим необслуживания.

При старте приложения терминал находится в режиме загрузки. В этот момент происходит инициализация компонентов приложения, а также настройка устройств.

Режим обслуживания служит для работы с клиентами и включает в себя два подрежима:

– подрежим ожидания клиента;

– подрежим транзакции.

Режим сервисного обслуживания и инкассации ВОТ служит для проверки состояния оборудования и проведения инкассации. Требует авторизацию.

Терминал переходит в режим необслуживания при нерабочем состоянии оборудования, потере связи с сервером и других непредвиденных ситуациях, а также перед перезагрузкой или выключением по запросу оператора.

Схема режимов работы ВОТ представлена на рисунке 6.3.



Рисунок 6.3 Схема режимов работы ВОТ

Описание условий перехода между режимами и подрежимами представлено в таблице 6.1.

Таблица 6.1 – Условия перехода между режимами

|  |  |
| --- | --- |
| № стрелки | Условие перехода |
| 1 | Все устройства подключены и готовы к работе. |
| 2 | 30 проверок готовности устройств дали отрицательный результат. |
| 3 | Купюра вставлена в купюроприемник. |
| 4 | Транзакция не подтверждена (отменена) пользователем. |
| Транзакция завершена успешно, клиент забрал деньги. |
| 5 | Произведен вход в сервисный режим (авторизация пройдена успешно). |

Продолжение таблицы 6.1

|  |  |
| --- | --- |
| 6 | Нажата кнопка «Выйти» в верхней строке интерфейса. |
| 7 | Одно или несколько устройств не готовы к работе. |
| Потеряна связь с сервером. |
| Ошибка при переинициализации купюроприемника при обновлении списка принимаемых банкнот. |
| Ошибка при смене курсов валют по запросу сервера. |
| 8 | Все устройства готовы к работе. |
| 9 | Устройства не готовы к работе либо потеряна связь с сервером при выходе из сервисного режима. |
| 10 | Произведен вход в сервисный режим (авторизация пройдена успешно). |
| 11 | Пропало питание. |
| Ошибка возврата банкноты при отмене транзакции пользователем. |
| Ошибка выдачи белорусских банкнот (банкнота успешно возвращена). |
| Ошибка возврата банкноты после неудачной попытки выдачи белорусских банкнот. |
| Пропала связь с сервером. |
| Неудачная попытка забрать и сбросить деньги, забытые пользователем. |

# РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Для правильной работы подсистемы на компьютере-сервере должно быть установлено следующее программное обеспечение:

– Операционная система из семейства Microsoft Windows 7/8/8.1;

– Microsoft Internet Information Services (IIS) 7.0 и выше;

– Microsoft SQL Server 2014 и выше;

– Microsoft .NET Framework 4.5 и выше.

Для установки ПС терминала на компьютер необходимо выполнить следующие действия:

1) скопировать на жесткий диск компьютера каталог программы TerminalApp;

2) выполнить скрипт setUserSettings.ps;

3) после перезагрузки системы выполнить скрипт configure.ps;

Для установки ПС сервера на компьютер необходимо выполнить следующие действия:

1) скопировать на жесткий диск компьютера каталог программы Server;

2) сконфигурировать виртуальный каталог с помощью оснастки IIS на основе скопированного ранее физического каталога;

3) подключить файл базы данных «atm.mdf» и файл журнала транзакций «atm\_log.ldf»;

4) настроить конфигурационный файл приложения «web.config» путем прописывания в его секции «connectionString» соответствующей строки подключения к серверу базы данных.

Для установки ПС бизнес администратора на компьютер необходимо выполнить следующие действия:

1) скопировать на жесткий диск компьютера каталог программы ArmUi;

2) сконфигурировать виртуальный каталог с помощью оснастки IIS на основе скопированного ранее физического каталога;

3) подключить файл базы данных «atm.mdf» и файл журнала транзакций «atm\_log.ldf»;

4) настроить конфигурационный файл приложения «web.config» путем прописывания в его секции «connectionString» соответствующей строки подключения к серверу базы данных.

Для работы с ПС бизнес администратора на клиентской машине нужно установить любой из Интернет-браузеров. При этом необходимо настроить браузер на возможность использования JavaScript.

## Работа с ПС терминала

### Ожидание пользователя

Интерфейс взаимодействия с пользователем в подрежиме ожидания отображает пользователю ВОТ следующую информацию:

– текущую дату и время;

– языковую версию интерфейсов взаимодействия с пользователем, установленную в ВОТ;

– курсы валют, действующие в ВОТ на текущий момент;

– номиналы и валюту принимаемые ВОТ к обмену;

– имя банка, предоставляющего услуги покупки валюты на ВОТ;

– рекламно-справочный блок, предоставленный банком.

Данный интерфейс предоставляет пользователю возможность выбора и изменения языковой версии интерфейсов взаимодействия с пользователем.

На рисунке 7.1 отображена страница ожидания клиента.

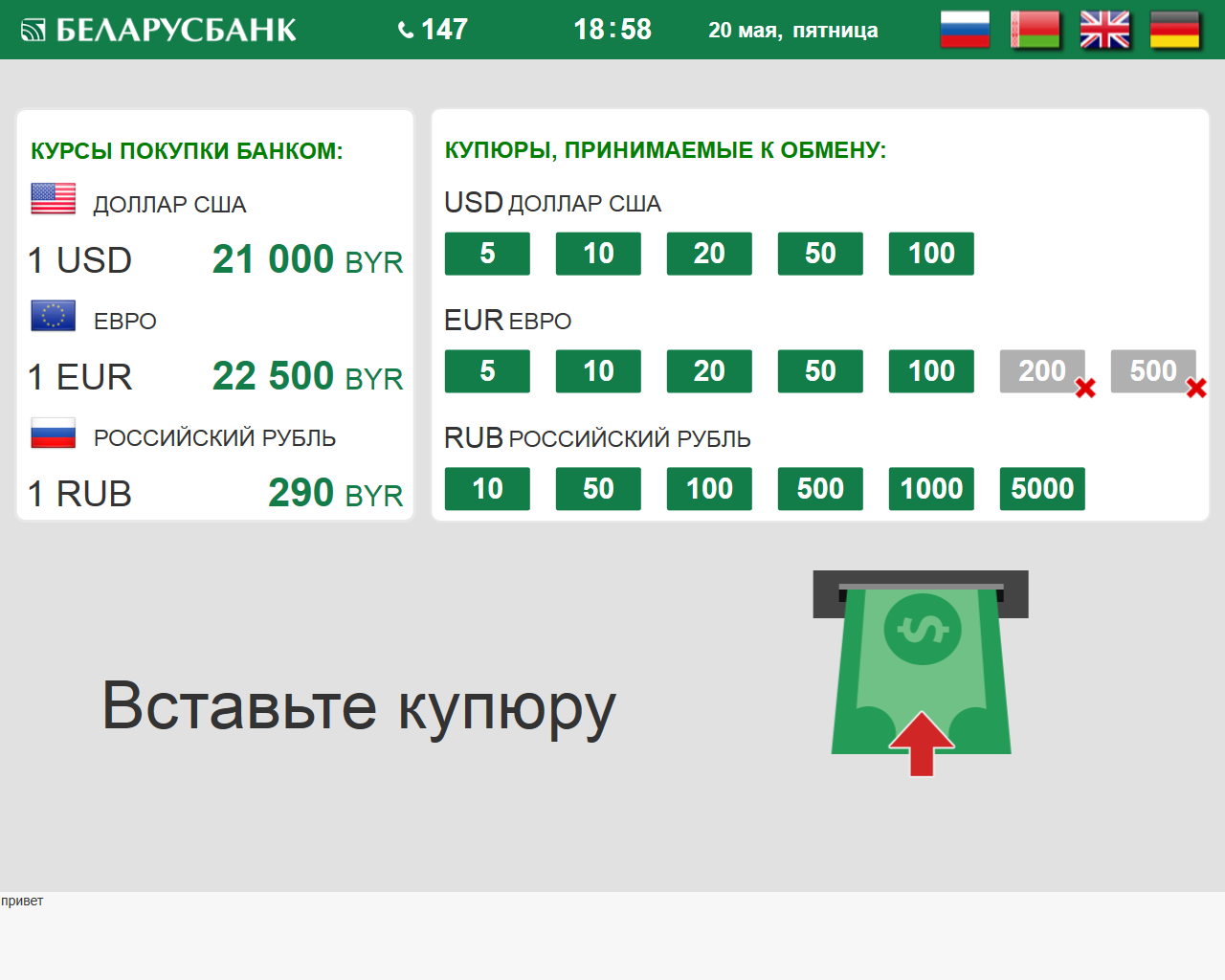


Рисунок 7.1 – Страница ожидания клиента

### Информация по транзакции

Вставка купюры в купюроприемник инициирует начало валютно-обменной операции. При этом, если вставленная купюра не распознана купюроприемником, либо данная купюра запрещена к приему на ВОТ, на экране появляется соответствующая информация, а вставленная купюра возвращается. На этом транзакция считается завершенной. Если вставленная купюра распознана и разрешена к приему, на экране ВОТ появляется страница информации о транзакции.

Данный интерфейс отображает пользователю информацию о ходе выполнения валютно-обменной операции, которую инициировал пользователь на ВОТ. Страница расчета по транзакции включает в себя:

– отображение пользователю результатов проверки купюры, вставленной им в купюроприемник ВОТ;

– отображение действий, которые пользователь должен выполнить в рамках текущей валютно-обменной операции для ее выполнения либо завершения/прекращения;

– отображение пользователю условий проведения валютно-обменной операции в ССВОО: номинал и валюта принятой к обмену купюры, установленный курс обмена для принятой купюры в ССВОО, сумма белорусских рублей, подлежащих выдаче;

– согласование с пользователем предлагаемых ему условий проведения валютно-обменной операции в ССВОО (возможность отказа от операции, подтверждение операции пользователем);

– отображение пользователю результатов выполнения валютно-обменной операции в ССВОО, которую он инициировал (отказ, подтверждение, программно-технические сбои).

Данный интерфейс предоставляет пользователю возможность:

– отказаться от проведения валютно-обменной операции на предлагаемых ему ССВОО условиях и получить купюру из купюроприемника;

– подтвердить проведение валютно-обменной операции на предлагаемых ему ССВОО условиях и получить белорусские рубли.

На рисунке 7.2 отображена страница информации по транзакции при распознанной купюре.

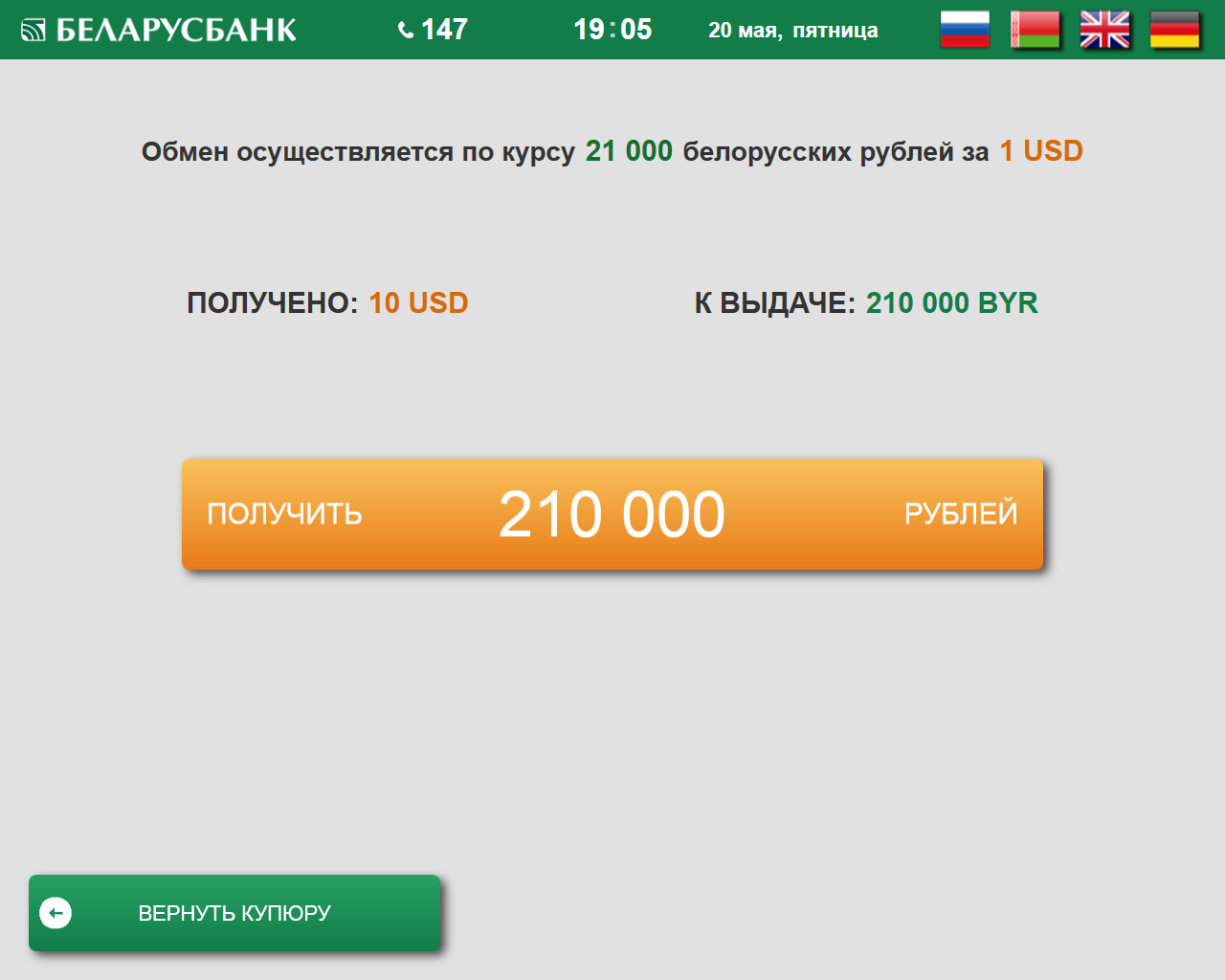


Рисунок 7.2 – Страница информации по транзакции

### Вход в сервисный режим

Отображение страницы входа в сервисный режим происходит при нажатии на «секретную кнопку», расположенную в хедере пользовательского интерфейса. Данный интерфейс отображает пользователю страницу со следующими элементами:

– поле для ввода имени пользователя;

– поле для ввода пароля;

– экранная клавиатура, содержащая цифробуквенные символы, символы переноса и подчеркивания, а также кнопку «Enter».

После нажатия на кнопку «Enter» происходит проверка введенных данных. В случае некорректных значений на экране отображается соответствующая информация. При успешной авторизации происходит вход в сервисный режим, после чего отображается страница с вкладками «Инкассация», «Оборудование» и «Кассеты».

На рисунке 7.3 приводится страница входа в сервисный режим.

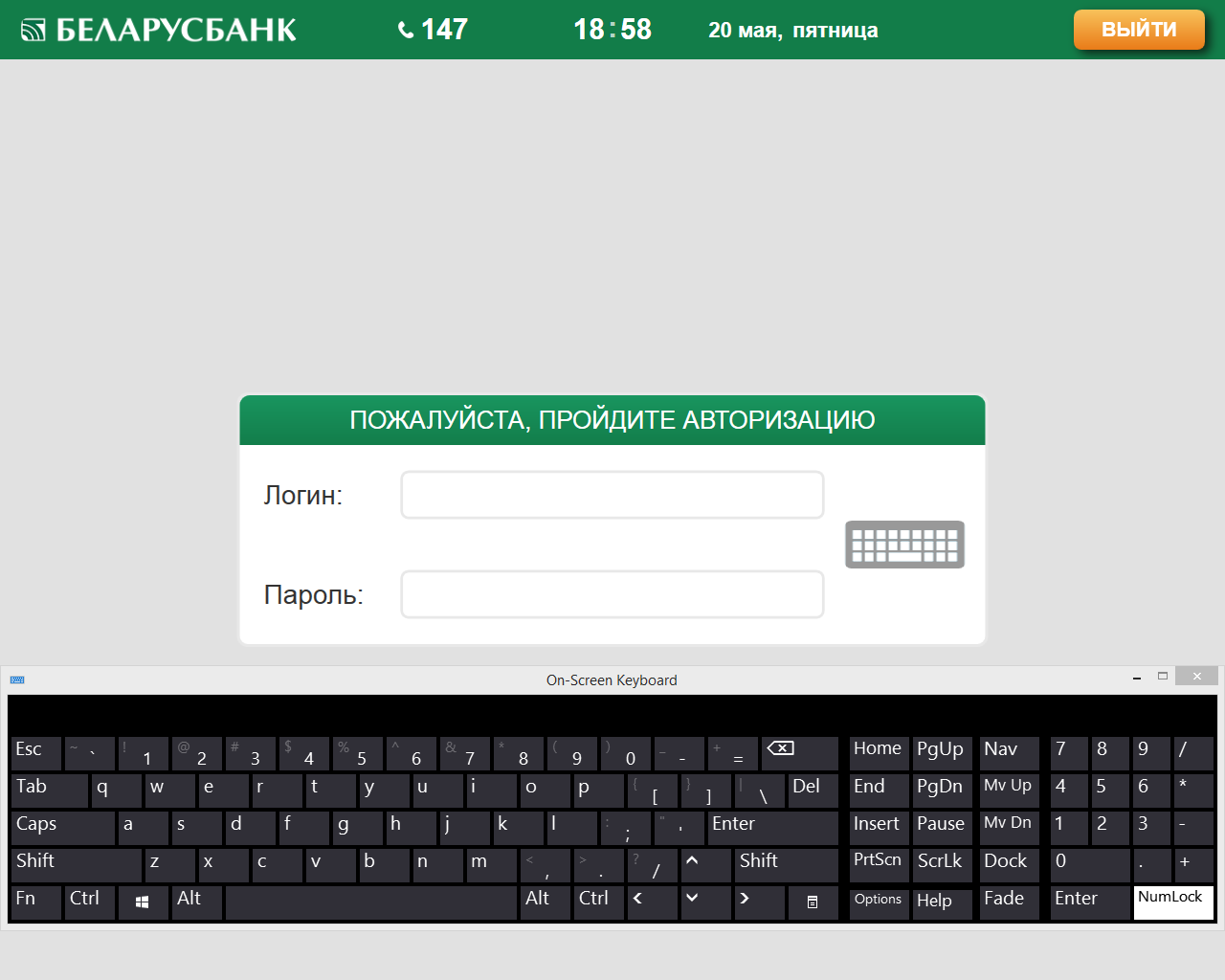


Рисунок 7.3 – Страница входа в сервисный режим

### Вкладка «Инкассация»

7.5.4.1 Печать значений счетчиков

Прежде чем начать инкассацию, необходимо выполнить печать счетчиков. Для этого необходимо нажать виртуальную кнопку «Печать счетчиков». По ее нажатии ВОТ выдаст чек с информационными данными о количестве принятых терминальным устройством купюр с указанием значений валют и разделением по номиналам каждой валюты с отображением общей суммы принятых денежных средств для каждой валюты, а также с информацией о количестве купюр, оставшихся в каждой кассете, в разрезе номиналов.

Страница печати счетчиков отображена на рисунке 7.4.

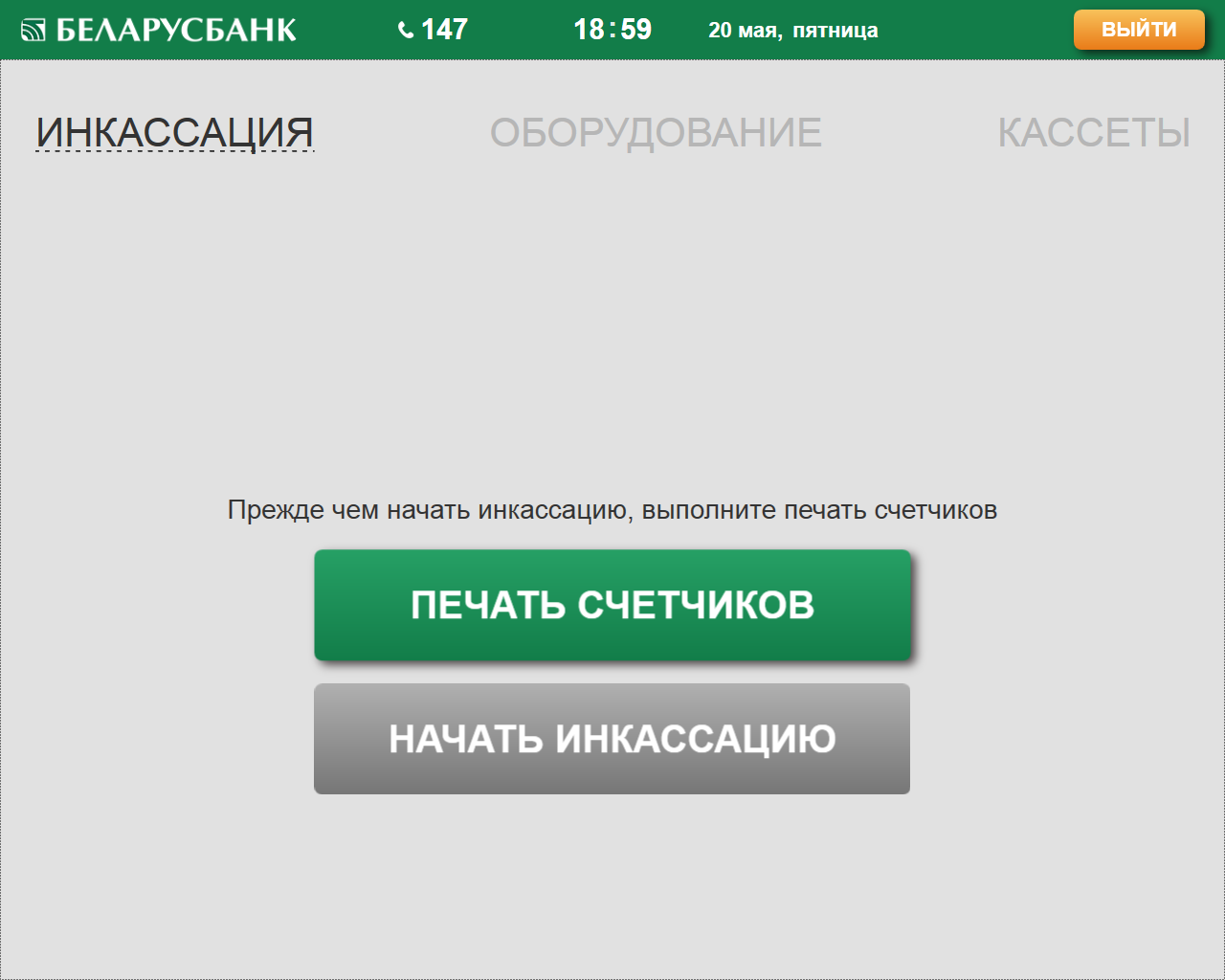


Рисунок 7.4 – Страница печати счетчиков

7.5.4.2 Процесс инкассации

Перед началом процесса инкассации необходимо выполнить печать счетчиков (п. 7.5.4.1.).

После успешного проведения данной операции кнопка «Начать инкассацию» становится активной.

Для разблокировки кассет и их замены необходимо нажать на сенсорную клавишу «Начать инкассацию».

При возникновении какой-либо ошибки управляющая программа ВОТ выдаст предупредительное сообщение «Не удалось разблокировать кассеты диспенсера». Иначе, на экране ВОТ отобразится меню изменения значений диспенсера со следующими функциональными элементами:

– записи кассеты с полями для установки значений номиналов и количества купюр;

– виртуальная клавиатура для ввода необходимых значений;

– функциональный элемент для завершения процесса инкассации «Завершить инкассацию».

Страница установки счетчиков кассет показана на рисунке 7.5.

После замены кассет необходимо установить значения номиналов и количество купюр в каждой записи кассеты.

Все поля должны быть обязательно заполнены. Если какие-либо кассеты являются пустыми, то для них необходимо ввести значение «0», используя виртуальную клавиатуру.

После заполнения данных по всем кассетам необходимо нажать виртуальную кнопку «Завершить инкассацию».

Если какие-либо поля не заполнены, появится информационное сообщение: «Заполните все данные по кассетам». Иначе, появляется сообщение об успешном завершении инкассации.

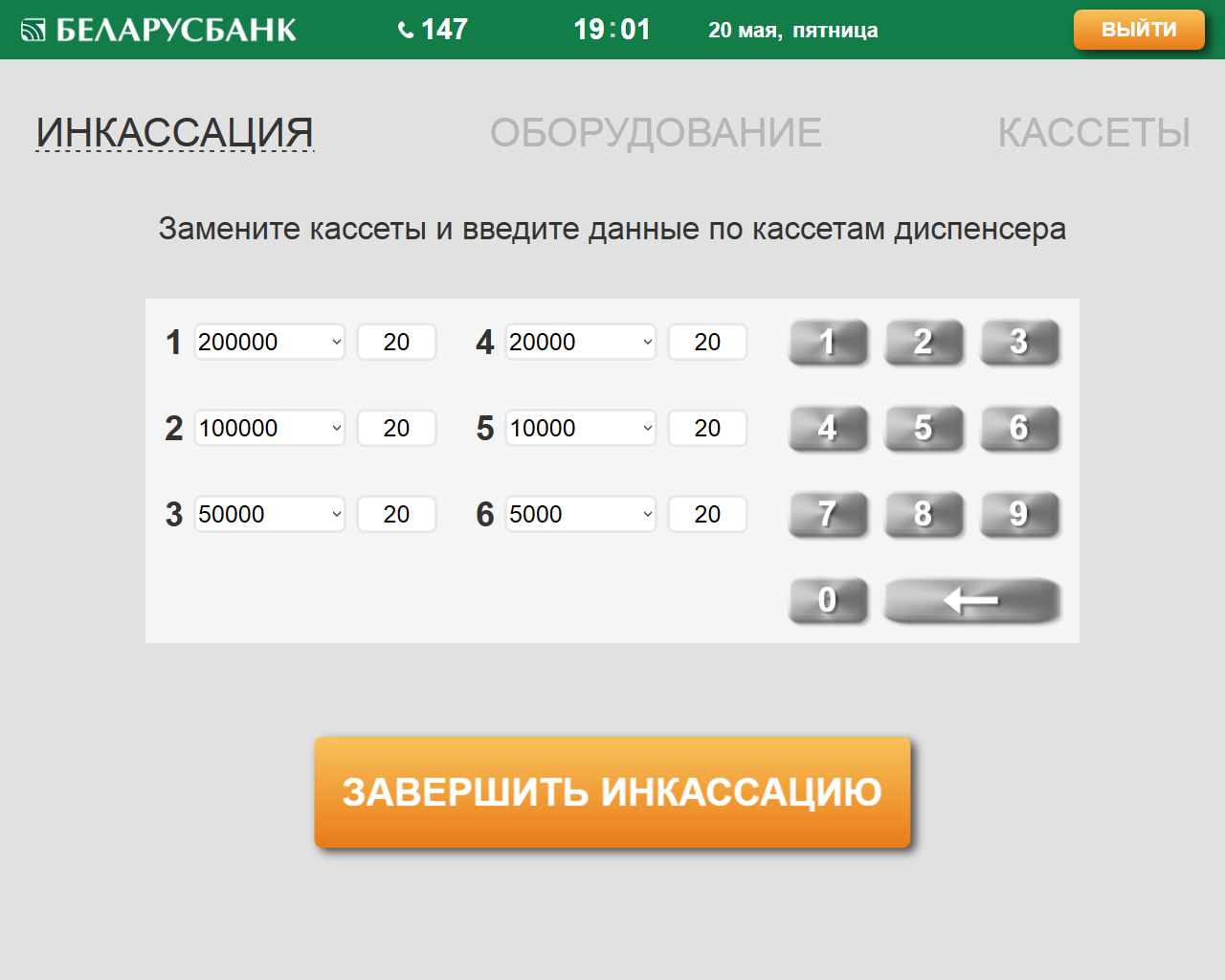


Рисунок 7.5 – Страница установки счетчиков кассет

### Вкладка «Оборудование»

Данная вкладка отображает следующие функциональные элементы:

– статусы работоспособности устройств;

– кнопка обновления статуса устройств «Обновить»;

– кнопка проверки диспенсера «Тестовый набор»;

– кнопка проверки купюроприемника «Тест купюроприемника»;

– кнопка проверки шаттера «Шаттер»;

– кнопка «Сброс диспенсера»;

– кнопка «Выгрузка»;

– кнопка перезагрузки терминала «Перезагрузка терминала»;

– кнопка выключения терминала «Выключить».

Рисунок 7.6 отображает вкладку «Оборудование».

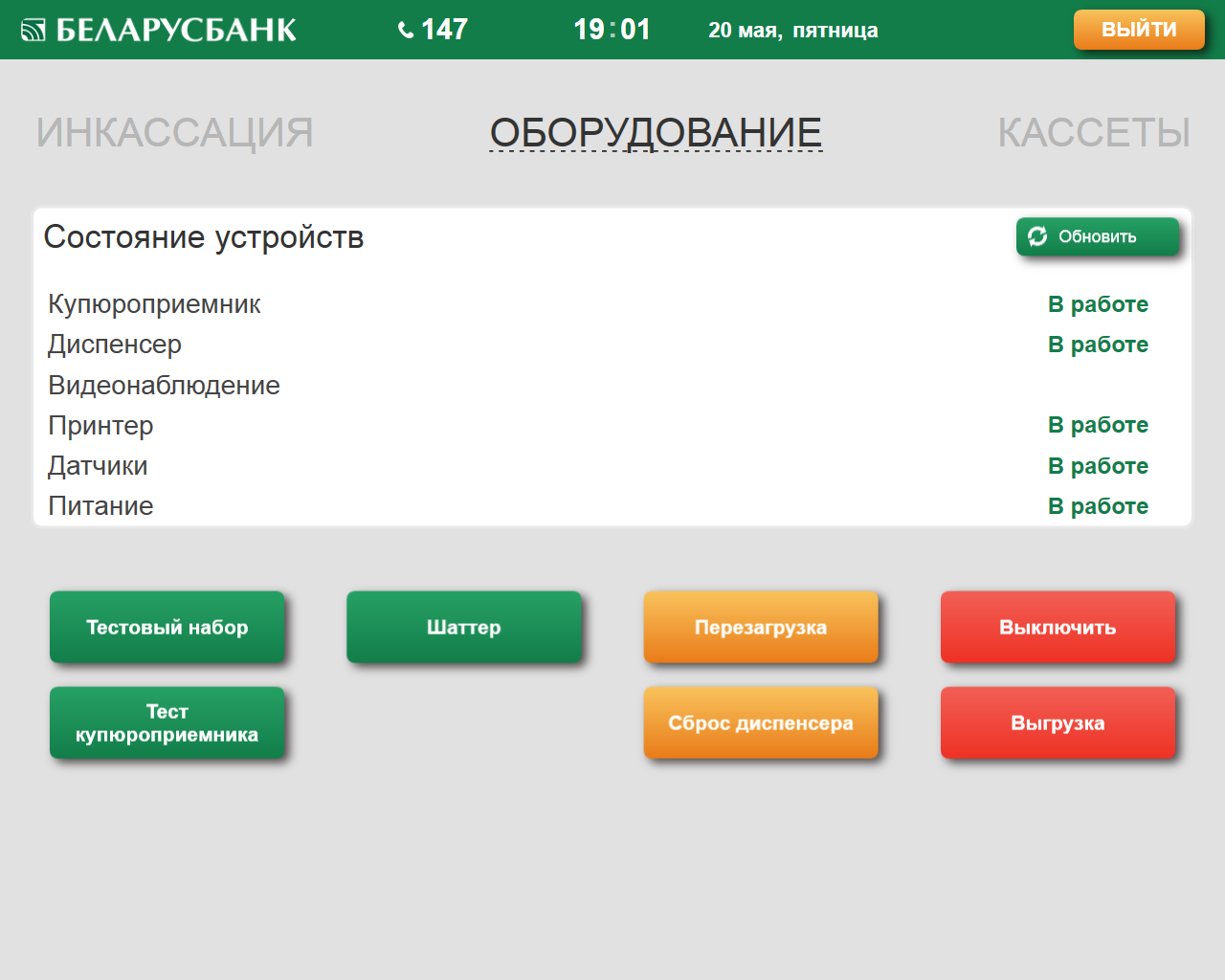


Рисунок 7.6 – Вкладка «Оборудование»

### Выход из сервисного режима

Для выхода из сервисного режима необходимо нажать на кнопку «Выйти» в правом верхнем углу экрана. ВОТ перейдет в режим ожидания клиента.

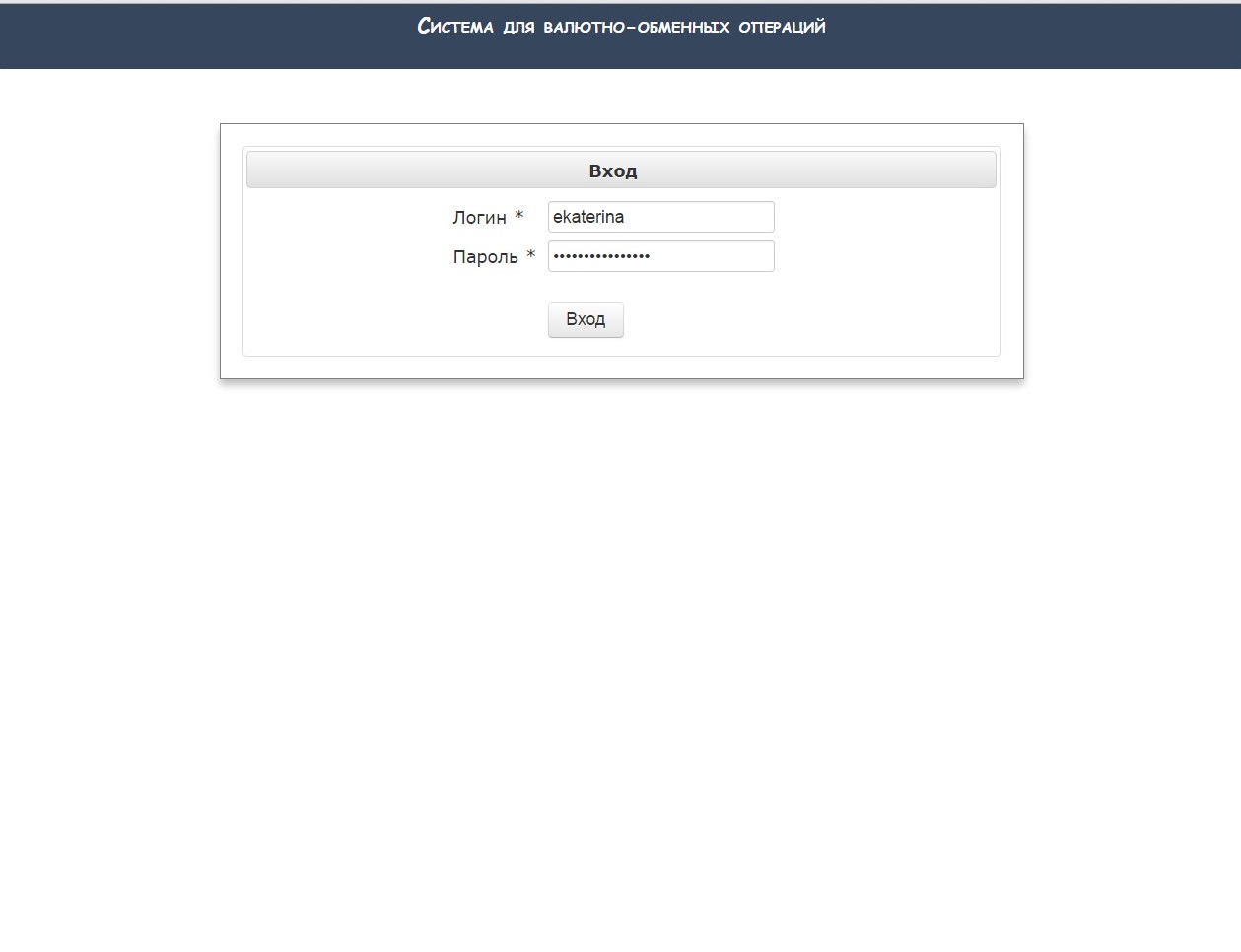
## Работа с ПС бизнес администратора

### Аутентификация

При загрузке веб-приложения первой открывается страница авторизации. Здесь пользователю нужно в соответствующих полях ввода ввести имя пользователя и пароль, а затем нажать кнопку «Вход».

Если пользователь вводит неверное значение имени или пароля, будет выведено соответствующее сообщение. Если имя и пароль правильные, пользователь будет перенаправлен на страницу со списком терминалов.

Страница авторизации отображена на рисунке 7.7.

Рисунок 7.7 – страница авторизации в АРМ

### Отображение списка терминалов

Страница со списком терминалов приводится на рисунке 7.8. На данной странице отображены все имеющиеся в базе данных терминалы, доступные для просмотра данному пользователю, включая номер терминала, имя его владельца, место установки, регион, текущие курсы валют, а также статус терминала. Также для каждого терминала доступны ссылки для просмотра детальной информации и удаления. Таблица поддерживает возможность сортировки по столбцам по возрастанию/убыванию. Для этого необходимо нажать на заголовок соответствующего столбца. Над таблицей находится форма для поиска терминала по номеру и владельцу, а также ссылка, которая перенаправит на страницу добавления нового терминала.

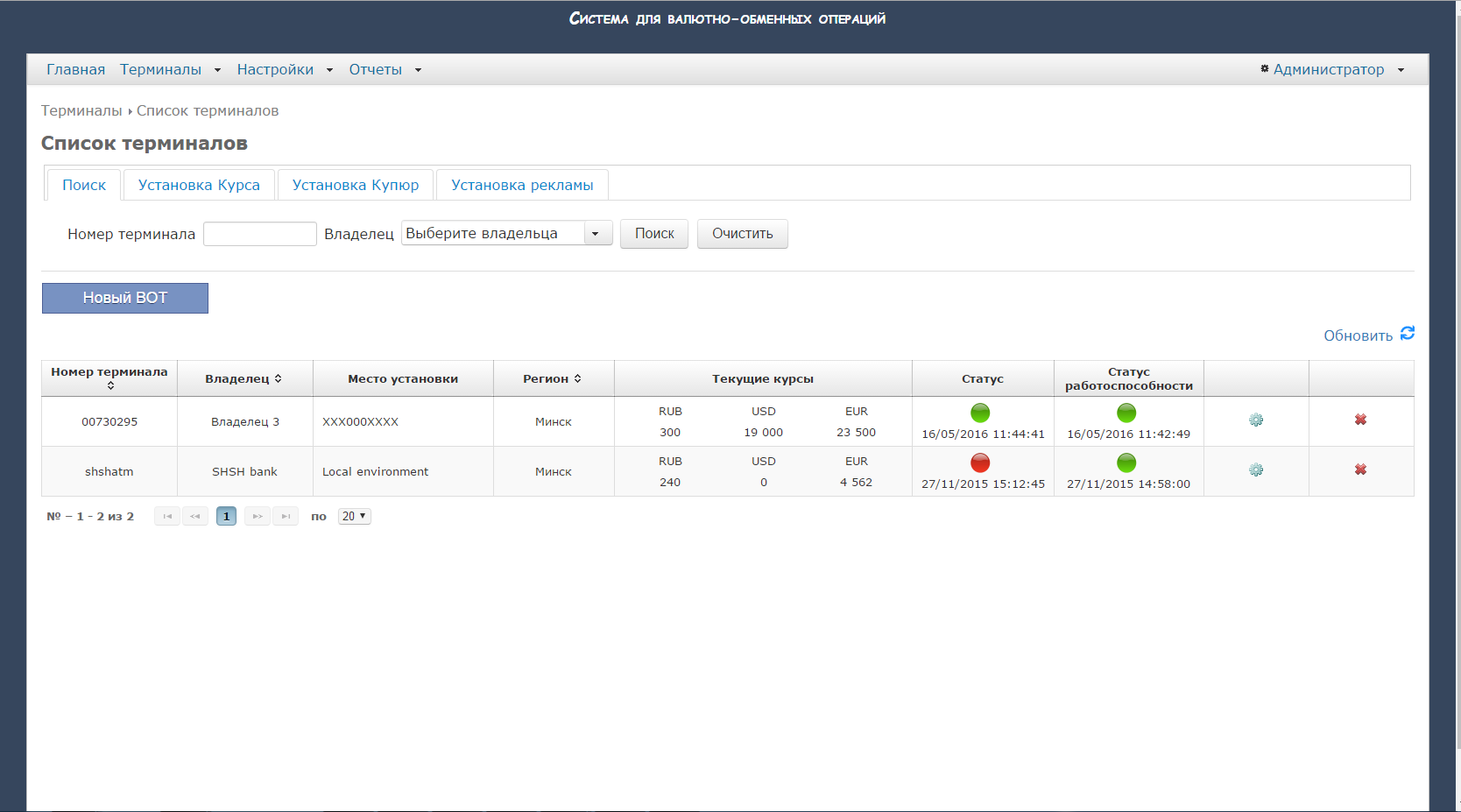


Рисунок 7.8 ­– Страница «Список терминалов»

### Добавление нового терминала

После нажатия на ссылку «Новый ВОТ» над таблицей будет отображена страница добавления нового терминала. На данной странице задается номер терминала, место его установки, его владелец и регион. Данная страница представлена на рисунке 7.9.

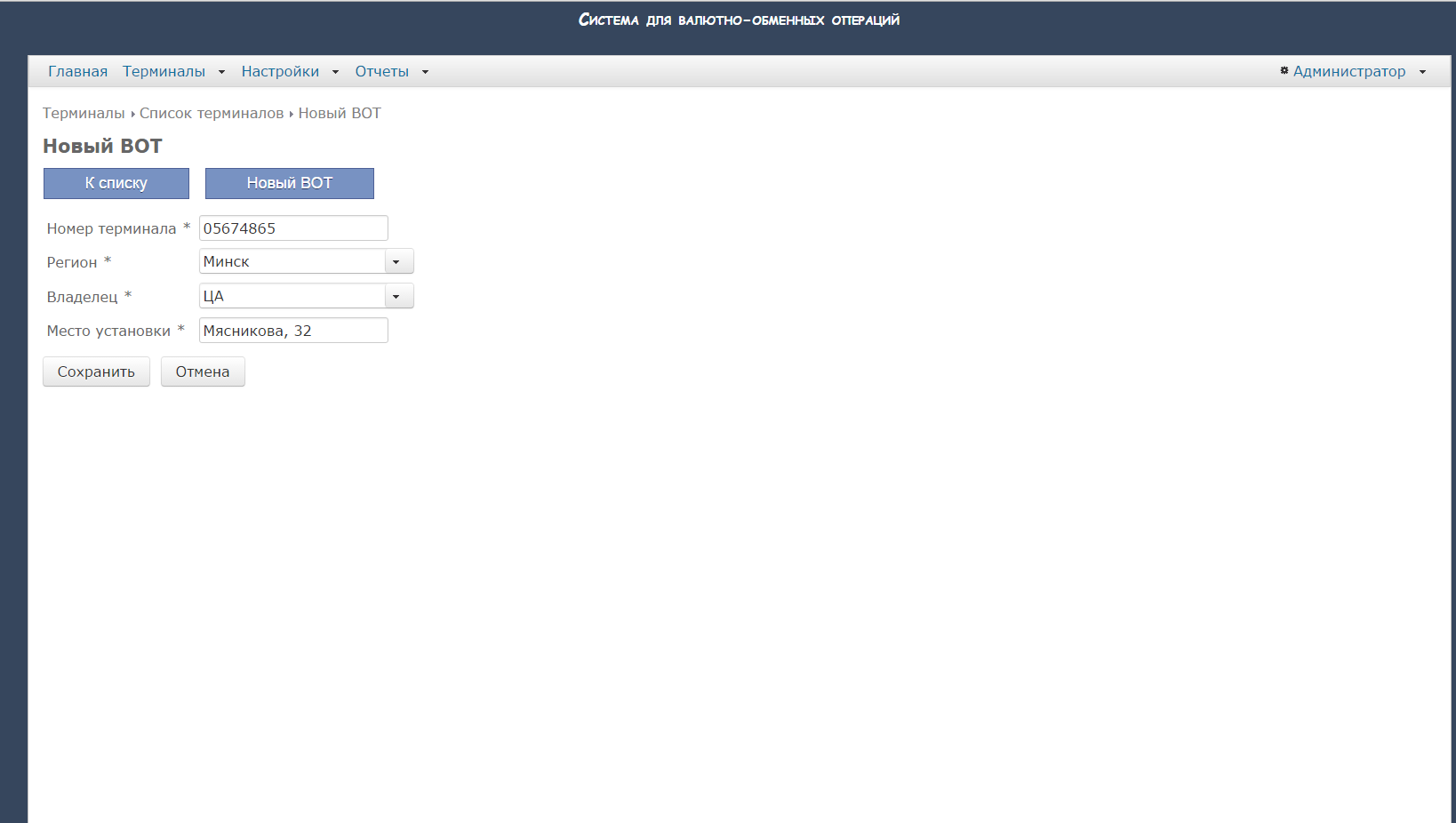


Рисунок 7.9 ­– Страница «Новый ВОТ»

При правильном задании всех параметров и нажатии на кнопку «Сохранить», пользователь вернется на страницу с отображением списка терминалов, который пополнится вновь созданным терминалом.

При нажатии на кнопку «Отменить» произойдет переход на страницу со списком терминалов.

### Просмотр детальной информации терминала

Для просмотра детальной информации о терминале необходимо кликнуть по ссылке в колонке «Детальная информация» в соответствующей строке таблицы. При этом произойдет переход на соответствующую страницу.

Данная страница содержит общую информацию о терминале, а также состояние оборудования и значения счетчиков купюроприемника и диспенсера. Вкладки «Установка Курса», «Установка Купюр» и «Установка рекламы» служат для настройки терминала. Вкладка «Валютные операции» содержит список транзакций, произведенных на терминале. На вкладках «События» и «Логи» находится информация о работе терминала. Вкладка «Настройка» служит для перезагрузки и выключения ВОТ.

Страница детальной информации о терминале изображена на рисунке 7.10.

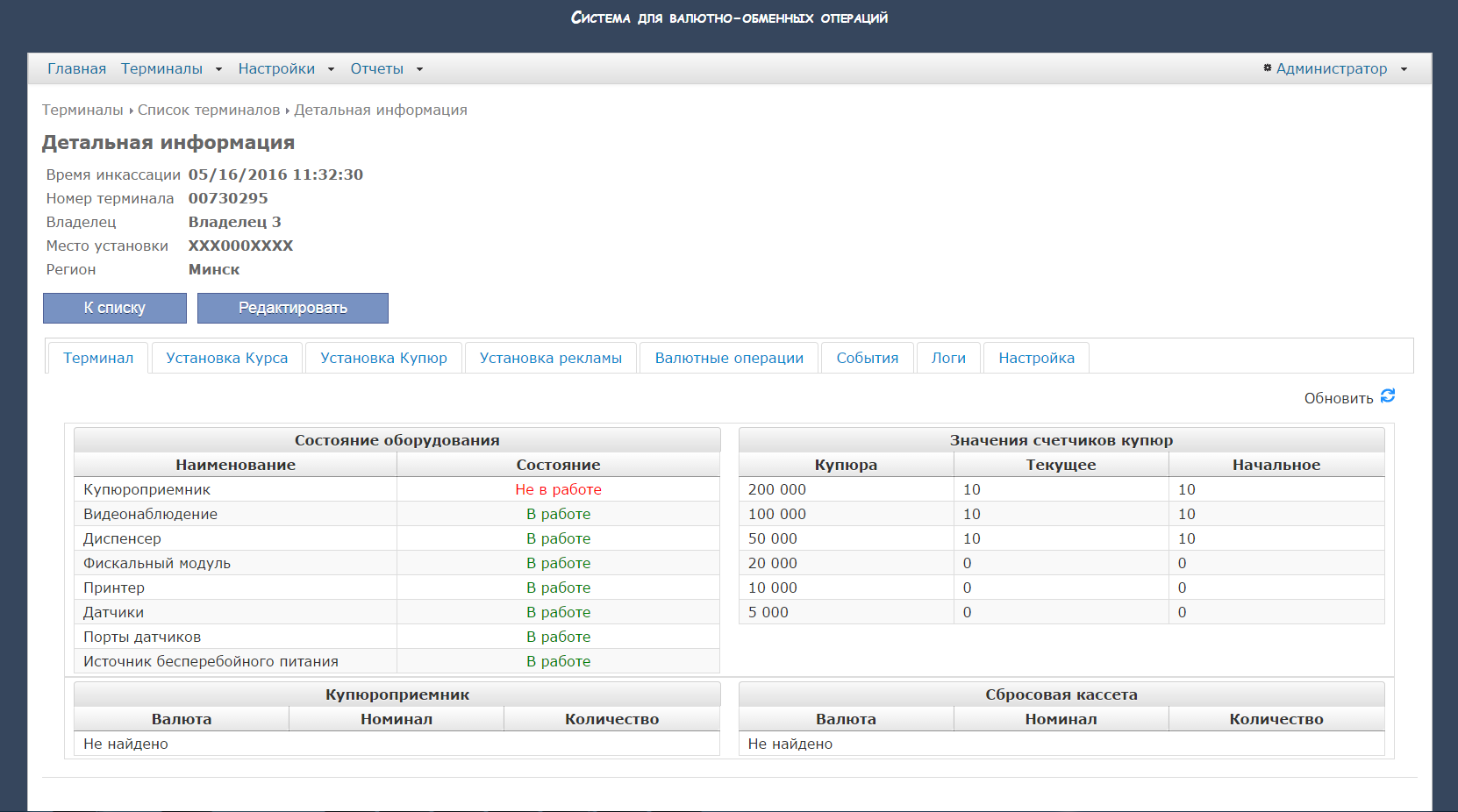


Рисунок 7.10 ­– Страница детальной информации терминала

### Установка курсов валют

Курсы валют можно устанавливать, как для одного терминала, так и для всех доступных терминалов. Для установки курсов валют для всех доступных терминалов необходимо перейти на вкладку «Установка курса» на странице со списком терминалов. При этом произойдет переход на страницу с формой установки курсов валют.

Данная форма содержит следующие параметры:

– валюта, для которой будет устанавливаться курс;

– числовое значение курса для выбранной валюты;

– дата и время установки (опционально), с которой будет действовать данный курс.

Страница установки курсов валют для всех доступных терминалов приводится на рисунке 7.11.

Установка курсов валют для конкретного терминала производится со вкладки «Установка курса» на странице детальной информации о терминале. Данная страница, помимо непосредственно формы установки курса, содержит информацию о текущих, а также запланированных курсах валют для выбранного терминала.

Страница установки курсов валют для конкретного терминала представлена на рисунке 7.12.

При верном заполнении данных полей и нажатии на кнопку «Установить курс», данные формы заносятся в базу данных, а пользователю отображается сообщение об успешной установке курса.

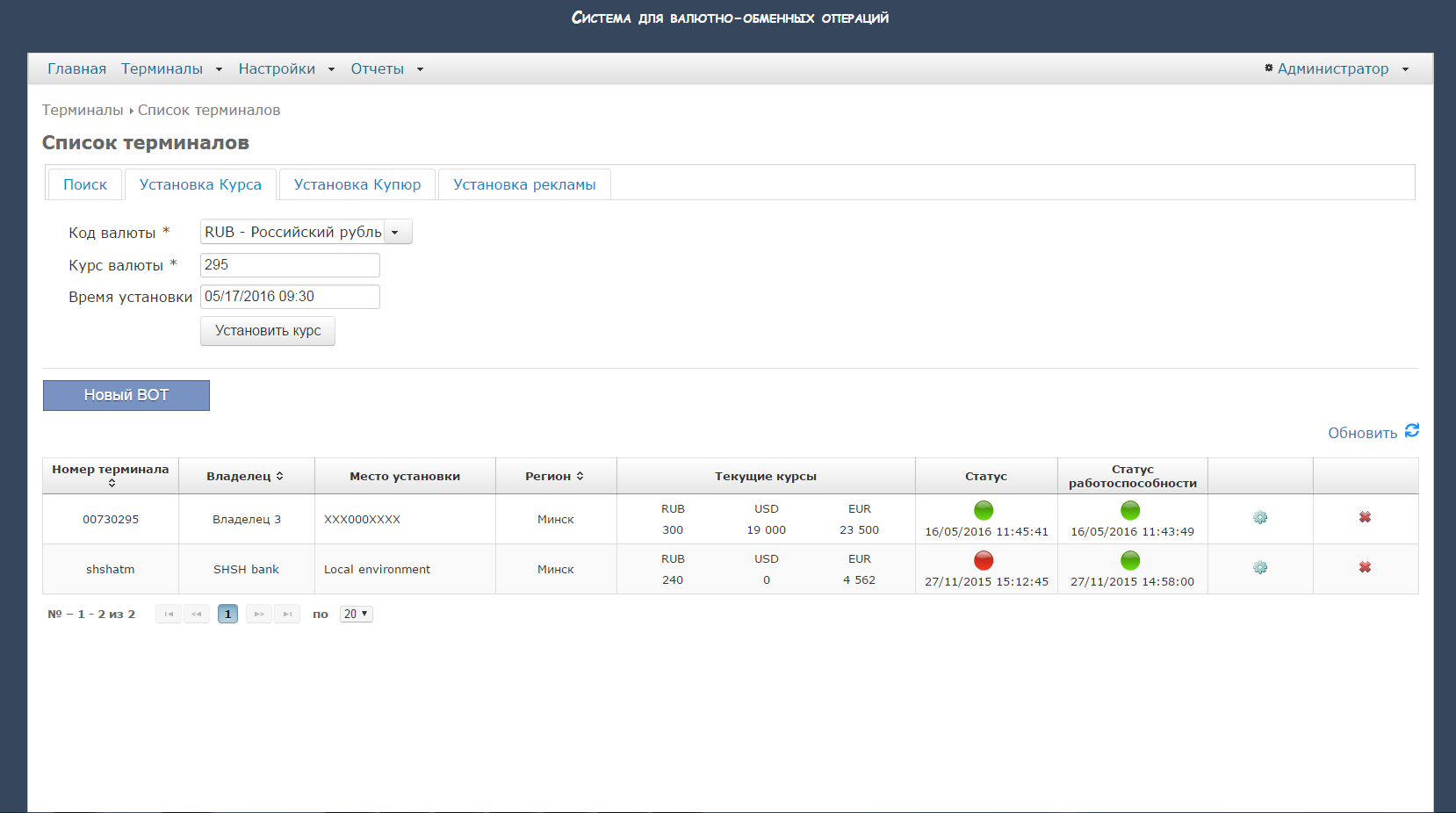


Рисунок 7.11 – Страница установки курсов валют

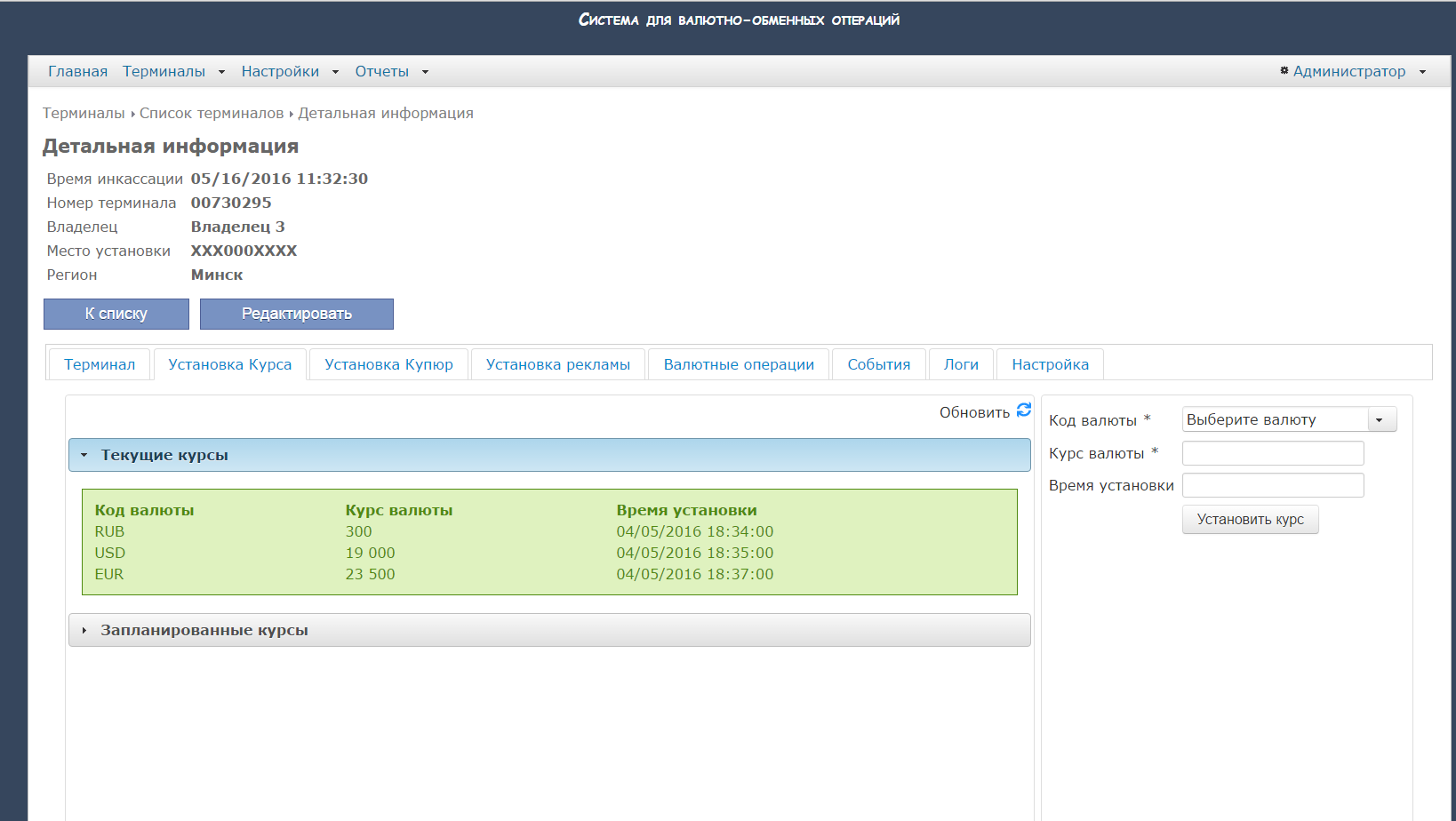


Рисунок 7.12 – Страница установки курсов валют для терминала

### Установка принимаемых банкнот

Установка купюр, доступных к обмену, также может производиться как для одного терминала, так и для группы терминалов. Для установки принимаемых банкнот для всех терминалов либо конкретного терминала необходимо перейти соответственно на вкладку «Установка купюр» на странице со списком терминалов либо на одноименную вкладку странице просмотра детальной информации о терминале. При этом произойдет переход на страницу с формой установки принимаемых банкнот. Данная форма содержит список всех доступных к приему банкнот в разрезе валют. Нажатием на валюту можно разрешать/запрещать ее к приему.

Отметив разрешенные и запрещенные к приему банкноты и нажатии на кнопку «Установить», данные о банкнотах сохраняются, а пользователю отображается сообщение об успешной установке принимаемых банкнот.

Страница установки принимаемых банкнот для всех доступных терминалов приводится на рисунке 7.13.

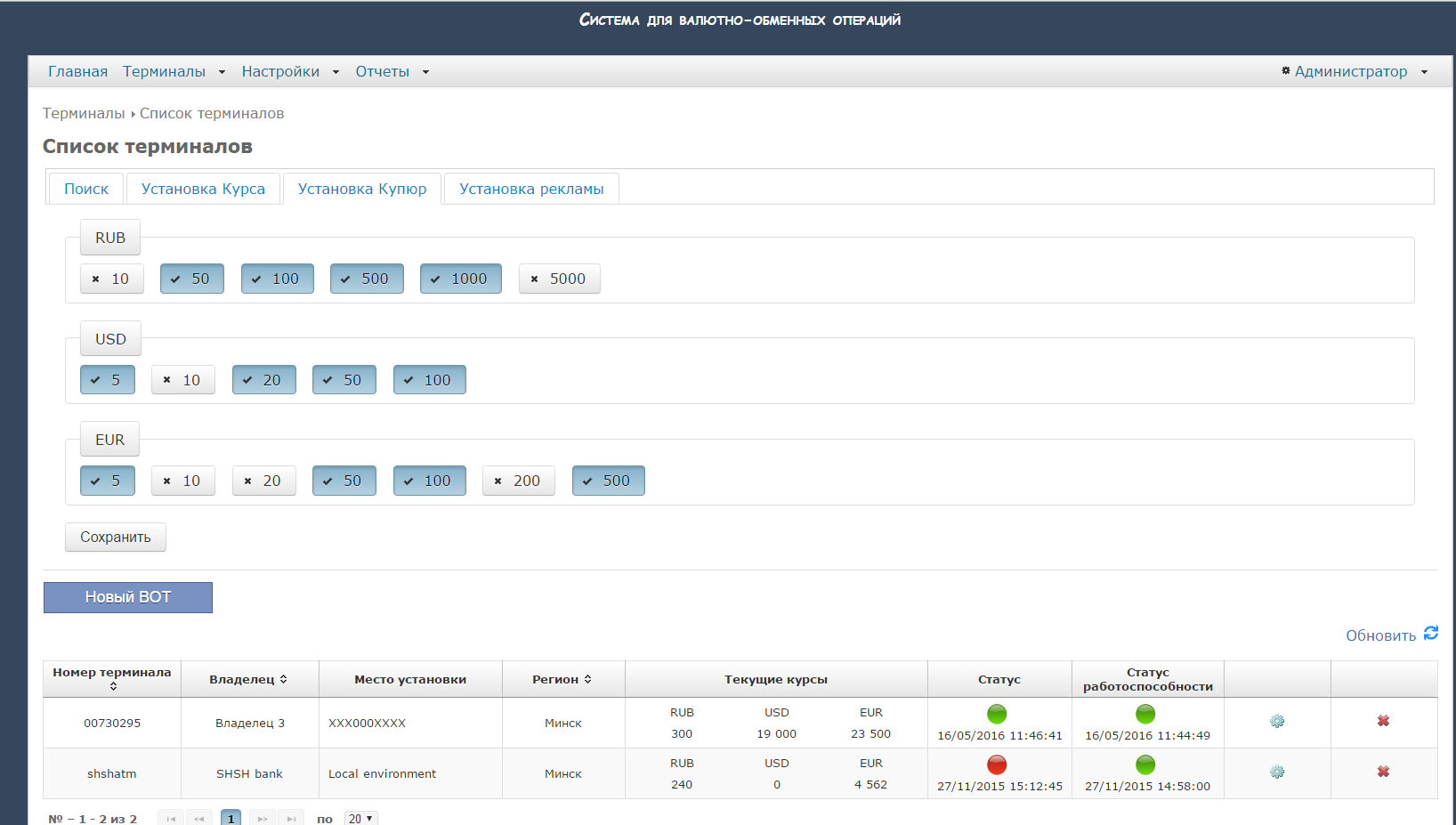


Рисунок 7.13 – Страница установки принимаемых банкнот

# ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ДИПЛОМНОГО ПРОЕКТА

## Введение и исходные данные

Целью технико-экономического обоснования ПП является определение экономической выгодности создания данного продукта и дальнейшего его применения.

Разработка клиентского терминала для обмена валют позволит создать устройство, позволяющее выполнять в режиме самообслуживания валютно-обменные операции с использованием наличных денежных средств.

Использование данного терминала позволит снизить временные затраты клиента на выполнение валютно-обменных операций за счет автоматизации процесса обмена валют. Для заказчика использование терминала значительно снизит издержки на содержание пунктов обмена валют.

Разработка терминала обмена валют предусматривает следующие стадии проектирования: техническое задание, технический проект, технорабочий проект, внедрение. Программный комплекс относится к 3-й группе сложности. Категория новизны продукта – В.

Для оценки экономической эффективности разработанного программного продукта рассчитаем смету затрат, цену программного продукта и прибыль.

## Расчет сметы затрат и цены программного продукта

Таблица 8.1 – Исходные данные

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование показателей | Буквенные обозначения | Единицы измерения | Количество |
| Коэффициент новизны | Кн | единиц | 0,7 |
| Группа сложности |  | единиц | 3 |
| Дополнительный коэффициент сложности | Кс | единиц | 1 |
| Поправочный коэффициент, учитывающий использование типовых программ | Кт | единиц | 0,7 |

Продолжение таблицы 8.1

|  |  |  |  |
| --- | --- | --- | --- |
| Установленная плановая продолжительность разработки | Тр | лет | 0,8 |
| Продолжительность рабочего дня | Тч | ч | 8 |
| Тарифная ставка 1-го разряда | Тм1 | тыс. руб. | 3765 |
| Коэффициент премирования | Кп | единиц | 1,3 |
| Норматив дополнительной заработной платы исполнителей | Нд | % | 15 |
| Отчисления в фонд социальной защиты населения | Зсз | % | 34 |
| Отчисления в Белгосстрах | Нне | % | 0,6 |
| Расходы на научные командировки | Рнк | % | 30 |
| Прочие прямые расходы | Пз | % | 20 |

Объем ПП определяется путем подбора аналогов на основании классификации типов ПП, каталога функций ПП и каталога аналогов ПП в разрезе функций, которые постоянно обновляются и утверждаются в установленном порядке. На основании информации о функциях разрабатываемого ПП по каталогу функций определяется объем функций. Объем ПП определяется на основе нормативных данных, приведенных в таблице 8.2.

Таблица 8.2 – Характеристика функций и их объем

|  |  |  |
| --- | --- | --- |
| Номер функции | Наименование (содержание) функций | Объем функций |
| 101 | Организация ввода информации | 150 |
| 102 | Контроль, предварительная обработка и ввод информации | 450 |
| 110 | Организация ввода/вывода информации с сети терминалов | 3200 |
| 111 | Управление вводом-выводом | 2400 |

Продолжение таблицы 8.2

|  |  |  |
| --- | --- | --- |
| 203 | Формирование баз данных | 2180 |
| 204 | Обработка наборов и записей базы данных | 1520 |
| 207 | Манипулирование данными | 9550 |
| 208 | Организация поиска и поиск в базе данных | 5480 |
| 501 | Монитор ПП (управление работой компонентов) | 740 |
| 502 | Монитор системы (управление работой комплекса ПП) | 7740 |
| 503 | Управление внешними устройствами и объектами | 5900 |
| 506 | Обработка ошибочных и сбойных ситуаций | 410 |
| 507 | Обеспечение интерфейса между компонентами | 970 |
| 605 | Вспомогательные и сервисные программы | 580 |
| 703 | Расчет показателей | 460 |
| 707 | Графический вывод результатов | 480 |
| Итого: | | 42210 |

Общий объем строк кода ПП рассчитывается по формуле:

|  |  |
| --- | --- |
|  | (8.1) |

где  *–* общий объем ПП;

– объем отдельной функции ПП;

n – общее число функций.

|  |  |
| --- | --- |
| . | (8.2) |

## Расчет нормативной трудоемкости

На основании общего объема ПП определяется нормативная трудоемкость (Тн) с учетом сложности ПП. Для ПП 3-й группы сложности, к которой относится разрабатываемый программный продукт, нормативная трудоемкость доставит 863 человеко-дня.

Коэффициент сложности, посредством которого учитываются дополнительные затраты труда, связанные со сложностью разрабатываемого программного продукта, рассчитывается по формуле:

|  |  |
| --- | --- |
|  | (8.3) |

где – коэффициент, соответствующий степени повышения сложности ПП за счет конкретной характеристики;

n – количество учитываемых характеристик.

|  |  |
| --- | --- |
|  | (8.4) |

На основании нормативной трудоемкости рассчитывается общая трудоемкость , чел./дн., с учетом распределения ее по стадиям:

|  |  |
| --- | --- |
|  | (8.5) |

где – трудоемкость разработки ПО на i-й стадии (чел./дн.);

n – количество стадий разработки.

Трудоемкость стадии ТЗ:

|  |  |
| --- | --- |
|  | (8.6) |

|  |  |
| --- | --- |
|  | (8.7) |

Трудоемкость стадии ЭП:

|  |  |
| --- | --- |
|  | (8.8) |

|  |  |
| --- | --- |
|  | (8.9) |

Трудоемкость стадии ТП:

|  |  |
| --- | --- |
|  | (8.10) |

|  |  |
| --- | --- |
|  | (8.11) |

Трудоемкость стадии РП:

|  |  |
| --- | --- |
|  | (8.12) |

|  |  |
| --- | --- |
|  | (8.13) |

Трудоемкость стадии ВН:

|  |  |
| --- | --- |
|  | (8.14) |

|  |  |
| --- | --- |
|  | (8.15) |

Общая трудоемкость определяется как сумма трудоемкостей по стадиям:

|  |  |
| --- | --- |
|  | (8.16) |

|  |  |
| --- | --- |
|  | (8.17) |

На основе уточненной трудоемкости разработки ПП и установленного периода разработки, общая плановая численность разработчиков равна:

|  |  |
| --- | --- |
|  | (8.18) |

где – плановая численность разработчиков (чел.);

– годовой эффективный фонд времени работы одного работника в течение года (дней в год);

– плановая продолжительность разработки ПП (лет).

|  |  |
| --- | --- |
|  | (8.19) |

Эффективный фонд времени работы в одного работника , дн./год:

|  |  |
| --- | --- |
|  | (8.20) |

где – количество дней в году;

– количество праздничных дней в году;

– количество выходных дней в году;

– количество дней отпуска.

|  |  |
| --- | --- |
| . | (8.21) |

При утверждении плановой численности разработчиков продолжительность разработки определяется по формуле:

|  |  |
| --- | --- |
|  | (8.22) |

где – срок разработки ПП (лет);

– трудоемкость разработки ПП на i-й стадии (человеко-дней);

– численность разработчиков ПП на i-й стадии (чел.);

– годовой эффективный фонд времени работы одного работника в течение года (дней в год);

m – число стадий.

|  |  |
| --- | --- |
|  | (8.23) |

Таблица 8.3 – Расчет трудоемкости ПП и численности исполнителей по стадиям

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Показатели | Стадии | | | | | Итого |
|  | ТЗ | ЭП | ТП | РП | ВН |  |
| Коэффициенты удельных весов трудоемкости стадии разработки ПО (d) | 0.09 | 0.07 | 0.07 | 0.61 | 0.16 | 1.0 |
| Распределение нормативной трудоемкости ПО (Тн) по стадиям, чел./дн. | 78 | 60 | 60 | 527 | 138 | 863 |
| Коэффициент сложности ПО (Кс) | 1 | 1 | 1 | 1 | 1 | – |

Продолжение таблицы 8.3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Коэффициент, учитывающий использование типовых программ () | – | – | – | 0.7 |  | – |
| Коэффициент, учитывающий новизну ПО (Кн) | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | – |
| Уточненная трудоемкость стадий (Ту), человеко-дней | 54 | 42 | 42 | 258 | 97 | 493 |
| Численность исполнителей (Чр), чел. | 3 | 3 | 3 | 3 | 3 | 3 |
| Сроки разработки, лет | 0,08 | 0,06 | 0,06 | 0,39 | 0,15 | 0,75 |

## Расчет основной заработной платы исполнителей

Реализацией проекта занимались три человека. В соответствии с численностью и выполняемыми функциями устанавливается штатное расписание группы специалистов-разработчиков.

Расчет основной заработной платы осуществляется в следующей последовательности.

Определим месячные (Тм) и часовые (Тч) тарифные ставки руководителя проекта (тарифный разряд – 16; тарифный коэффициент – 3,72), инженера-программиста I категории (тарифный разряд – 14, тарифный коэффициент – 3,25) и инженера-программиста (тарифный разряд – 12, тарифный коэффициент – 2,84).

Месячная тарифная ставка каждого исполнителя (Тм) определяется путем умножения действующей месячной тарифной ставки 1-го разряда (Тм1) на тарифный коэффициент (Тк), соответствующий установленному тарифному разряду:

|  |  |
| --- | --- |
|  | (8.24) |
|  |  |

Часовая тарифная ставка рассчитывается путем деления месячной тарифной ставки на установленный при пятидневной рабочей неделе и восьмичасовом рабочем дне среднемесячный фонд рабочего времени:

|  |  |
| --- | --- |
|  | (8.25) |

где – среднемесячный фонд рабочего времени

Определим тарифные ставки руководителя проекта:

|  |  |
| --- | --- |
|  | (8.26) |

|  |  |
| --- | --- |
|  | (8.27) |

Месячная и часовая тарифные ставки инженера-программиста I категории равны соответственно:

|  |  |
| --- | --- |
|  | (8.28) |

|  |  |
| --- | --- |
|  | (8.29) |

Месячная и часовая тарифные ставки инженера-программиста:

|  |  |
| --- | --- |
|  | (8.30) |

|  |  |
| --- | --- |
|  | (8.31) |

Расчет месячных и часовых тарифных ставок приведен в таблице 8.4.

Основная заработная плата исполнителей рассчитывается по формуле:

|  |  |
| --- | --- |
|  | (8.32) |

где n – количество исполнителей;

– часовая тарифная ставка i-го исполнителя (тыс. руб.);

– количество часов работы в день;

– эффективный фонд рабочего времени i-го исполнителя (дней);

– коэффициент премирования.

Таблица 8.4 – Штатное расписание группы разработчиков

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Должность | Количество ставок | Тарифный разряд | Тарифный коэффициент | Месячная тарифная ставка (тыс. руб.) | Часовая тарифная ставка (тыс. руб.) |
| Руководитель проекта | 1,00 | 16 | 3,72 |  |  |
| Инженер-программист I категории | 1,00 | 14 | 3,25 |  |  |
| Инженер-программист | 1,00 | 12 | 2,84 |  |  |

|  |  |
| --- | --- |
|  | (8.33) |

Следовательно, сумма основной заработной платы составляет:

|  |  |
| --- | --- |
| 82,48 · 8 · 158,4 1,3 + 72,06 8 158,4 1,3 + 62,97 8 158,4 = 254813,68 тыс. руб. = 254 813 680 руб. | (8.34) |

Дополнительная заработная плата () определяется по нормативу в процентах к основной заработной плате по формуле:

|  |  |
| --- | --- |
|  | (8.35) |

где – норматив дополнительной заработной платы в целом по научной организации, равный 20%.

Дополнительная заработная плата () составляет:

|  |  |
| --- | --- |
|  | (8.36) |
|  |  |

Таблица 8.5 – Расчет себестоимости и отпускной цены ПП

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование статей | Норматив | Методика расчета | Значение, тыс. руб. |
| Отчисления в фонд социальной защиты |  |  | 103963,98 |
| Отчисления в Белгосстрах |  |  | 1834,66 |
| Материалы и комплектующие | / 100 строк |  | 173,06 |
| Спецоборудование | − |  | 30000 |
| Машинное время | − | машино-часов | 4389,84 |
| Расходы на научные командировки |  |  | 76444,10 |
| Прочие прямые расходы |  |  | 50962,74 |
| Накладные расходы |  |  | 254813,68 |
| Полная себестоимость | – |  | 522582,06 |
| Прогнозируемая прибыль |  |  | 130645,52 |
| Прогнозируемая цена без налогов | – |  | 653227,58 |
| Налог на добавленную стоимость |  |  | 130645,52 |

Продолжение таблицы 8.5

|  |  |  |  |
| --- | --- | --- | --- |
| Отчисления налога в местный и республиканский бюджеты |  |  | 26509,76 |
| Прогнозируемая отпускная цена | – |  | 810382,86 |
| Освоение ПП |  |  | 52258,21 |
| Сопровождение ПП |  |  | 104516,41 |

## Оценка экономической эффективности применения ПП у пользователя

Для расчета экономического эффекта применения нового ПП необходимы данные имеющегося внедренного на производстве аналога (базового варианта). Некоторые из показателей базового варианта не могут быть получены (составляют коммерческую тайну либо защищены авторскими правами разработчиков). Поэтому расчет экономического эффекта будем проводить, опираясь на известные данные показателей базового и нового варианта ПП. Показатели обоих вариантов приведены в таблице 8.6.

Таблица 8.6 – Исходные данные для расчета экономического эффекта

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование показателей | Обозначения | Единицы измерения | Значение показателя | |
| базовый вариант | новый вариант |
| Капитальные вложения, включая стоимость услуг по сопровождению и адаптации ПП |  | тыс. руб. | – | 810382,86 |
| Затраты на освоение ПП |  | тыс. руб. | – | 52258,21 |
| Затраты на сопровождение ПП |  | тыс. руб. | – | 104516,41 |

Продолжение таблицы 8.6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Затраты на укомплектование ВТ техническими средствами в связи с внедрением нового ПП |  | тыс. руб. | – | 30000,00 |
| Затраты на пополнение оборотных фондов, связанных с эксплуатацией нового ПП |  | тыс. руб. | – | 100,00 |
| *Всего затрат:* | | | | |
| Время простоя сервиса, обусловленное ПО, в день | , | мин | 200 | 10 |
| Стоимость одного часа простоя |  | тыс. руб. | 600 | 600 |
| Среднемесячная зарплата одного программиста |  | тыс. руб. | 11300 | 11300 |
| Коэффициент начислений на зарплату |  | – | 2,1 | 2,1 |
| Среднемесячное количество рабочих дней |  | день | 18,33 | 18,33 |
| Количество типовых задач, решаемых за год | , | задача | 22000 | 22000 |
| Объем выполняемых работ | , | задача | 22000 | 22000 |
| Средняя трудоемкость работ в расчете на задачу | , | человеко-часов на задачу | 0,13 | 0,01 |
| Количество часов работы в день |  | час | 8 | 8 |
| Ставка налога на прибыль |  | % | 18 | 18 |

## 

## Расчет капитальных затрат

Общие капитальные вложения () заказчика (потребителя), связанные с приобритением, внедрением и использованием ПП, включают в себя затраты на доукомплектацию техническими средствами в связи с внедрением нового ПП, а также затраты на пополнение оборотных средств в связи с использованием нового ПП.

Таблица 8.7 – Расчет капитальных затрат

|  |  |  |
| --- | --- | --- |
| Наименование | Методика расчета | Значение (тыс. руб.) |
| Затраты пользователя на приобретение ПП по отпускной цене разработчика |  | 810382,86 |
| Затраты на освоение ПП |  | 52258,21 |
| Затраты на сопровождение ПП |  | 104516,41 |
| Затраты на пополнение оборотных средств связи с использованием нового ПП |  | 100,00 |
| Общие капитальные вложения |  | 967257,48 |
| Экономия затрат на заработную плату в расчете на 1 задачу |  | 9,25 |
| Экономия на заработную плату при использовании нового ПП |  | 203500 |
| Экономия с учетом начисления на заработную плату |  | 427350 |
| Экономия за счет сокращения простоя сервиса |  | 418000 |
| Общая годовая экономия текущих затрат, связанных с использованием нового ПП |  | 845350 |

Внедрение нового ПП позволит пользователю сэкономить на текущих затратах 845350 тыс. руб.

## Расчет экономического эффекта

Для пользователя в качестве экономического эффекта выступает лишь чистая прибыль – дополнительная прибыль, остающаяся в его распоряжении, которая определяется по формуле:

|  |  |
| --- | --- |
|  | (8.37) |

где – ставка налога на прибыль, равная 18%.

Тогда чистая прибыль будет равна:

|  |  |
| --- | --- |
| . | (8.38) |

В процессе использования нового ПП чистая прибыль в конечном итоге возмещает капитальные затраты. Однако полученные при этом суммы результатов (прибыли) и затрат (капитальных вложений) по годам приводят к единому времени – расчетному году (за расчетный год принят 2016 год) путем умножения результатов и затрат на каждый год на коэффициент приведения (), который рассчитывается по формуле:

|  |  |
| --- | --- |
|  | (8.39) |

где – норматив дисконтирования разновременных затрат и результатов;

– расчетный год, = 1;

t – номер года, результаты и затраты которого приводятся к расчетному (2016 – 1; 2017 – 2; 2018 – 3; 2019 – 4).

Коэффициентам приведения () по годам будут соответствовать следующие значения:

|  |  |
| --- | --- |
|  | (8.40) |

Таблица 8.8 – Расчет экономического эффекта от использования нового программного средства

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Показатели | Ед. измерения | 2016 | 2017 | 2018 | 2019 |
| *Результаты:* |  |  |  |  |  |
| Прирост прибыли за счет экономии затрат () | тыс. руб. | – | 693187 | 693187 | 693187 |

Продолжение таблицы 8.8

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Прирост прибыли с учетом фактора времени | тыс. руб. | – | 603073 | 524049 | 456117 |
| Затраты: |  |  |  |  |  |
| Приобретения ПП () | тыс. руб. | 810383 | – | – | – |
| Освоение ПП () | тыс. руб. | 52258 | – | – | – |
| Сопровождение ПП | тыс. руб. | 104516 | – | – | – |
| Пополнение оборотных средств () | тыс. руб. | 100,00 | – | – | – |
| Всего затрат | тыс. руб. | 967257 | – | – | – |
| То же с учетом фактора времени | тыс. руб. | 967257 | – | – | – |
| *Экономический эффект:* |  |  |  |  |  |
| Превышение результата над затратами | тыс. руб. | -967257 | 603073 | 524049 | 456117 |
| То же с нарастающим итогом | тыс. руб. | -967257 | -364184 | 159865 | 615982 |
| Коэффициент приведения | единиц | 1,000 | 0,870 | 0,756 | 0,658 |

## Выводы по технико-экономическому обоснованию

В результате технико-экономического обоснования было выявлено, что разработка ПО клиентского терминала для обмена валют является эффективной и выгодной для заказчика.

Чистая прибыль от реализации ПП ( = 693187 тысяч рублей) представляет собой экономический эффект от создания нового ПП.

Положительный экономический эффект достигается за счет уменьшения трудоемкости работ пользователей в расчете на одну задачу и уменьшения времени простоя системы.

Разработка данного программного продукта окупается менее, чем за два года, следовательно, является экономически целесообразной.

# ЗАКЛЮЧЕНИЕ

В данном дипломном проекте разработано программное обеспечение клиентского терминала для обмена валют. Для этого был проведен обзор литературы по теме, проанализированы аналогичные программные продукты, построены логическая и физическая модели данных, спроектированы и реализованы база данных, серверная и клиентские части системы, рассчитана экономическая эффективность.

Разработанная система предоставляет следующие возможности:

1) для клиента – обмен иностранных купюр на белорусские рубли;

2) для бизнес-администратора – контроль над системой, включающий просмотр данных по терминалам и валютно-обменным операциям, а также возможности управления курсами валют и принимаемыми банкнотами;

3) для инкассатора – управление значениями счетчиков кассет ВОТ;

4) дружественный и интуитивно понятный пользовательский интерфейс;

5) контроль над целостностью, корректностью и непротиворечивостью вводимых данных, а также возможности предотвратить попытку ввода некорректных данных;

6) возможность легкого сопровождения, а также дальнейшего расширения и наращивания функциональности;

Разработанная система обеспечивает:

1) для клиента – упрощение механизма и сокращение времени обмена иностранных купюр на белорусские рубли;

2) снижение издержек на содержание пунктов обмена валют;

3) уменьшение временных затрат на обслуживание клиентов;

4) круглосуточный режим обмена валют;

5) повышение качества обслуживания клиентов.

В ходе выполнения тестирования программный комплекс показал стабильные результаты работы. Было установлено, что система успешно работает в стандартном режиме эксплуатации. Таким образом, поставленная задача выполнена в полном объеме.

Такая система вполне способна выполнить возложенные на нее требования по обмену валют. Областью возможного практического применения является банковская сфера. Так образом, данный программный продукт является востребованным. Повсеместная автоматизация бизнес процессов и очень активное внедрение информационных технологий обеспечивают перспективность данной разработки.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Конноли Т., Бегг К., Страчан А. Базы данных: проектирование, реализация и сопровождение. Теория и практика, 2-е изд.: Пер. с англ.: Уч.пос. – М. Издательский дом «Вильямс», 2000. – 1120 с.: ил.

[2] Rose для разработчиков и ради разработчиков. – Режим доступа: www.citforum.ru/programming/application/rrose2.shtml, свободный - Загл. с экрана. – Яз. рус.

[3] Анализ и проектирование. Визуальное моделирование (UML) Rational Rose. – Режим доступа: www.info-system.ru/designing/methodology/rational/rational\_visual\_ model.htm, свободный - Загл. с экрана. – Яз. рус.

[4] Рамодин Д. Купи себе немножечко CASE. – Режим доступа: www.caseclub.ru/articles/rose1.html, свободный. – Загл. с экрана. – Яз. рус.,англ.

[5] Стандарт предприятия. Дипломные проекты (работы). Общие требования СТП01–2010. – Минск: БГУИР, 2011.

[6] Технико-экономическое обоснование дипломных проектов: Метод. пособие для студ. всех спец. БГУИР. В 4-х ч. Ч. 4: Проекты программного обеспечения/ В.А. Палицын. – Минск: БГУИР, 2006. – 76 с.