**Федеральное государственное образовательное бюджетное учреждение  
высшего образования**

**«Финансовый университет при Правительстве Российской Федерации»**

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

**ПМ.02 Разработка, администрирование и защита баз данных**

**Группа:3ПКС-215**

**УТВЕРЖДАЮ**

**Председатель цикловой комиссии**

**программирования и баз данных**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ПестовА.И/ \_\_.\_\_.2018**

**ПРОЕКТ КУРСОВОЙ**

**На тему: «Интеграция платформы 1С:Предприятие со сторонним программным обеспечением»**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**Руководитель курсового проекта**

**\_\_\_\_\_\_\_\_\_\_\_\_\_/Костиков П.А/**

**Исполнитель курсового проекта**

**\_\_\_\_\_\_\_\_\_\_\_\_\_ /Трипольский П.П./**

**Оценка за проект: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_.\_\_\_\_.2018г.**

2018

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc514626766)

[1. ОБЩИЙ РАЗДЕЛ 6](#_Toc514626767)

[1.1. Системные требования 6](#_Toc514626768)

[1.2. Программные требования 7](#_Toc514626769)

[2. ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ 8](#_Toc514626770)

[2.1. Предпроектное исследование предметной области 8](#_Toc514626771)

[2.1.1 Выбор способа взаимодействия платформы 1С:Предприятие с сторонними программными средствами 8](#_Toc514626772)

[2.1.2 Выбор средства для создания программы, с которой платформе 1С:Предприятие предстоит взаимодействовать 9](#_Toc514626773)

[2.2. Анализ требований и определение спецификаций программного обеспечения 14](#_Toc514626774)

[2.2.1 Концептуальная схема межпрограммного взаимодействия 14](#_Toc514626775)

[2.2.2 Описание бизнес-логики разрабатываемой конфигурации 16](#_Toc514626776)

[2.3. Проектирование программного обеспечения 17](#_Toc514626777)

[2.4 Разработка пользовательских интерфейсов программного обеспечения 21](#_Toc514626778)

[2.5. Организация защиты базы данных 24](#_Toc514626779)

[2.6. Тестирование и отладка программного обеспечения 24](#_Toc514626780)

[3. РУКОВОДСТВО ПО ИСПОЛЬЗОВАНИЮ ПРОГРАММЫ 25](#_Toc514626781)

[3.1. Руководство администратора 25](#_Toc514626782)

[3.1.1 Общие сведения о программе 25](#_Toc514626783)

[3.1.2 Настройка Firewall 25](#_Toc514626784)

[3.1.3 Инструкция по развертыванию информационной базы 1С 28](#_Toc514626785)

[3.1.4 Интеграция модуля считывания NFC карт в новые конфигурации 29](#_Toc514626786)

[3.2. Руководство пользователя 30](#_Toc514626787)

[3.2.1 Считывание номера метки 30](#_Toc514626788)

[3.2.2 Ведение учета 31](#_Toc514626789)

[ЗАКЛЮЧЕНИЕ 32](#_Toc514626790)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 33](#_Toc514626791)

[ПРИЛОЖЕНИЕ А 34](#_Toc514626792)

# ВВЕДЕНИЕ

На текущий момент на отечественном рынке программного обеспечения всё большую популярность набирают решения, работающие с платформой 1С:Предприятие. Вызвано это восстановлением собственной рыночной экономики и повышением спроса на системы учета для более эффективного ведения бизнеса и уплаты налогов. Как следствие – повышается спрос на программистов, умеющих работать с этой платформой.

Тем не менее, программная платформа 1С:Предприятие не только позволяет оперативно вести учет информации, но и предоставляет универсальное стандартизированное средство построения пользовательских интерфейсов. Эта особенность, в свою очередь, позволяет тратить меньше времени на обучение сотрудников взаимодействию с новым программным обеспечением: все программы на платформе 1C:Предприятие интуитивно понятны, так как логика визуальных компонентов дублируется. Например, человек, один раз работавший с программой на этой платформе, например, с продуктом бухгалтерского учета “1С:Бухгалтерия”, будет знать, что записи можно пометить на удаление. А удаляются записи крупными блоками из специального меню.

Как следствие – остро встает вопрос о организации взаимодействия программ на платформе 1С:Предприятие с другими средствами, так как язык программирования логики платформы не включает в себя всё множество взаимодействия с аппаратной частью компьютера: это не системный язык программирования, а высокоуровневый и интерпретируемый построчно в процессе исполнения. Платформа отлично взаимодействует с пользователем, хранит данные, но организовать на ней работу с внешними устройствами без создания и интеграции сторонней программы невозможно, так как она на это не рассчитана.

В этой работе я произвожу исследование средств взаимодействия базы данных на платформе 1С:Предприятие с внешними устройствами на примере создания базы данных контрольно пропускного пункта охраны колледжа, где студенты прикладывают условную кампусную карту для входа, чтобы избежать задержек в связи с ручной проверкой документов. В качестве условной кампусной карты будут использоваться транспортные карты “Тройка”, используемые организацией “Московский Транспорт”. Увидеть подобную карту можно на рисунке 1. Как результат будет создана конфигурация базы данных 1С:Предприятие, работающая с специальной программой, устанавливаемой отдельно и работающей с данными картами через “считыватель”.

  
Рисунок 1. Карта “Тройка”

# ОБЩИЙ РАЗДЕЛ

Данное программное обеспечение разрабатывалось и запускалось на устройствах со следующими характеристиками и требует их выполнения в комплексе:

## 1.1. Системные требования

Электронно-вычислительная машина:

* Процессор: Intel Core i9-7980XE;
* Оперативная память: 64 ГБ;
* Клавиатура и мышь.
* Монитор 1600х900;
* Свободное пространство на жестком диске 1 ТБ.

Считыватель:

* Мобильный телефон на базе ОС Android
* Поддержка модуля NFC
* Частота процессора более: 2000 MHz;
* Оперативная память: 8 ГБ;
* Свободное пространство в памяти устройства 5 ГБ.

Сеть:

* Устройства должны быть в одной сети Wi-Fi с пропускной способностью не менее 1000 Мбит/с для 2,4 ГГц и каналом 11.

## 1.2. Программные требования

Электронно-вычислительная машина:

* Операционная система Windows 10 Pro, сборка 1709
* База данных 1С:Предприятие 8.3
* Установленный .Net Framework 4.5
* DNS-суффикс локальной сети равный “lan”.
* Отключенный Windows Defender, Windows Updater и Windows Firewall для простоты настройки взаимодействия с внешними устройствами.

Считыватель:

* ОС Android 6.0. с разрешенной установкой из сторонних источников
* Включенный модуль NFC

Сеть:

* OpenWrt дистрибутив без какого либо контроля над локальной сетью.
* Адрес хостов локальной сети должен начинаться с 192.

# 2. ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ

## 2.1. Предпроектное исследование предметной области

### 2.1.1 Выбор способа взаимодействия платформы 1С:Предприятие с сторонними программными средствами

Существует несколько способов взаимодействия платформы 1С:Предприятие с сторонними программными средствами. В результате исследования я выяснил, что самыми популярными является клиент-серверное взаимодействие по протоколу HTTP и технология COM. Их сравнение приведено на рисунке 2.1.1. В результате сравнения был выбран протокол HTTP.

|  |  |  |
| --- | --- | --- |
| **Предмет сравнения** | **Технология взаимодействия** | |
| HTTP | COM |
| Свободность реализации | Свободная спецификация | Технологический стандарт Microsoft, реализация работает только в ОС семейства Windows NT |
| Кроссплатформенность | Да | Нет |
| Что можно передать? | Любые данные, потоковая передача | Указатель на метод, примитивные данные. Строго одно возвращаемое значение |
| Третьи лица в соединение | Непосредственное клиент-серверное соединие | Соединение через программные средства Microsoft Windows |
| Обработка ошибок | Легко определить и обработать | Небезопасный код |
| Богатый выбор языков программирования | Почти любые языки программирования | Только компилируемые и характерные для Microsoft Windows |

Рисунок 2.1.1 Сравнение средств межпрограммного взаимодействия

### 2.1.2 Выбор средства для создания программы, с которой платформе 1С:Предприятие предстоит взаимодействовать

Так как был выбран универсальный способ взаимодействия в виде реализации HTTP сервера, будут анализироваться лишь кроссплатформенные программные средства, покрывающие все ОС, где работает платформа 1С:Предприятие.

* **NodeJS**

Пример реализации HTTP-сервера

var http = require('http');

var url = require('url');

http.createServer(function (req, res) {

res.writeHead(200, {'Content-Type': 'text/html'});

var q = url.parse(req.url, true).query;

if(typeof q.Code != 'undefined'){

res.end("Результат запроса");

}

else{

res.end("Метод не найден");

}

}).listen(8080);

Данный язык программирования асинхронный и функциональный, обычно его используют для создания микро сервисов и небольших программ. Переменные вне области видимости функции являются глобальными и доступны по всей программе. Как следствие – не возможность ввода из консоли, что делает его узкоспециализированным и неприменимым в решении поставленной задачи на стороне сервера – нам нужно не только реализовать программу, но и создать пример для других студентов. Кроме того, пользовательский интерфейс для этого языка программирования пишется на разметке HTML, а она, в свою очередь потребует массы усилий для создания красивой и универсальной верстки. А ещё её нужно где-то запускать, это потребует запуск ещё одного процесса и усложнения программы.

* **С++ c библиотекой Qt5**

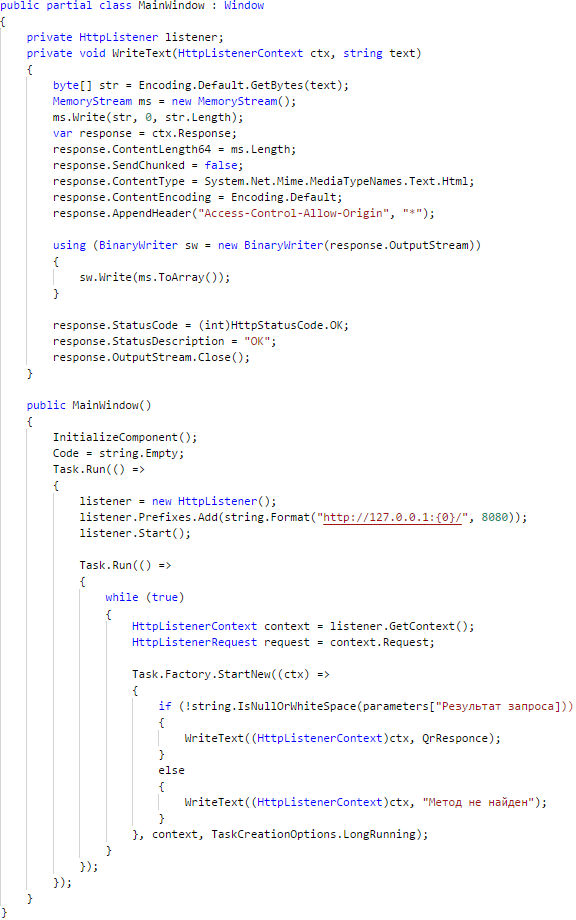
Пример кода на языке программирования C++ представлен ниже



Безусловно, данный язык программирования можно использовать, но только в сочетании с последними стандартами C++11 и библиотекой Qt. Без последних стандартов код, который и так громоздкий, становится начинающему разработчику просто нечитаемым. Кроме того, стандартная библиотека STL не умеет работать с сокетами, реализовать на ней HTTP сервер нельзя, можно только создав решения, ориентированные под конечную операционную систему. Это противоречит поставленной задаче – сделать код как можно более универсальным. А ещё, так как компания Nokia, руководящая разработкой библиотеки Qt развалилась, ожидать родную поддержку новой архитектуры процессоров не стоит. Как следствие, на отечественных процессорах это решение может не заработать. Помимо этого существует проблема компиляции и дистрибуции программы. Так как библиотека Qt разделена на модули, поставляемые под разными лицензиями, легко нарушить авторские права. На текущий момент в Российской Федерации действует Статья 146 УК РФ – Нарушение авторских и смежных прав. Эта статья защищает присвоение авторства, обязывая соблюдать лицензию поставки библиотеки, которая в свою очередь, требует или открытие исходных кодов или платной подписки. Купить библиотеку единоразовым платежом на момент 20 марта 2018 года нельзя, платная подписка за месяц стоит как минимальный размер оклада труда гражданина в Москве. Публиковать исходные коды программы не всегда возможно, так как конечное право владельца передается заказчику, а у программиста остаются лишь авторские права. В случае с государственными предприятиями, публикация исходных кодов является угрозой информационной безопасности, так как это позволит применить альтернативный метод поиска уязвимости – анализ исходных кодов. Этот способ позволит злоумышленнику запустить копию программы у себя дома и избежать журналирования, как следствие – сбор пакета документов и ответной реакции отдела информационной безопасности предприятия.

* **Язык программирования C#**

Пример кода на языке программирования C# представлен ниже

****

Видно, что код программы на языке программирования C# меньше кода на C++. Помимо этого, при программировании на C# можно применить множество дополнений сторонних разработчиков, выложенных в общественное пользование под лицензией MIT в пакетный менеджер NuGET. Это, в свою очередь, позволяет оперативно найти недостающий компонент и установить, что увеличивает возможности языка. На рисунке 2.1.2 представлена работа с NuGET в среде разработчика Microsoft Visual Studio

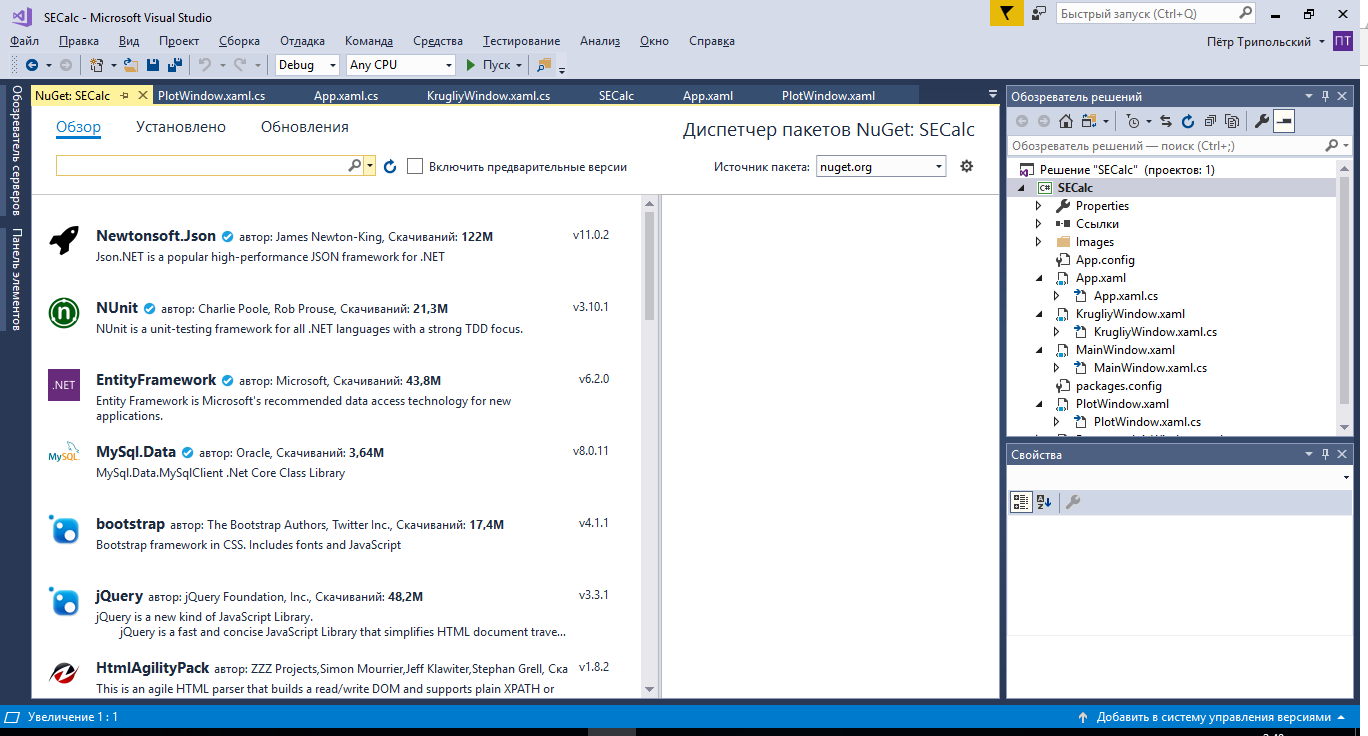


Рисунок 2.1.2. Работа с пакетным менеджером Nuget

Лицензия MIT позволяет поставлять программу с закрытыми исходными кодами, что делает язык программирования C# перспективнее C++ в разработке закрытых решений государственных предприятий. Кроме того, в отличие от Qt, в пакетном менеджере NuGET множество дополнений постоянно дополняется, а на конечный компьютер устанавливаются только нужные дополнения.

В качестве средства разработки программного обеспечения для считывателя меток, выраженного мобильным устройством на ОС Android, был выбран фреймворк Apache Callback, ныне известный как Adobe PhoneGap, в связи с отсутствием кроссплатформенных альтернатив.

## 2.2. Анализ требований и определение спецификаций программного обеспечения

### 2.2.1 Концептуальная схема межпрограммного взаимодействия

Данная реализация информационной системы разделена на три независимо работающих программных модуля, взаимодействующих по протоколу HTTP в режиме запрос-ответ. Платформа 1С:Предприятие будет запускать модуль, включающий в себя HTTP сервер, разработанный на языке программирования C#. После того, как будет произведен системный вызов запуска программы платформа дождется старта HTTP сервера и будет отправлять запросы с определенным интервалом, ответ на которые должен содержать отчет о результате работы третьего программного модуля, выраженного программой на ОС Android. Он, в свою очередь, будет получать данные о IP адресе компьютера в локальной сети через Qr код в диалоговом окне программы-сервера и отправлять запрос, содержащий данные из NFC метки. Сервер же, на полученный запрос всегда дает положительный ответ и запоминает данные. При следующем запросе базы данных 1С:Предприятие ей будет передано содержимое NFC метки, полученное из запроса на предыдущем шаге, а программа-сервер закроется. Ознакомиться с примерной блок-схемой взаимодействия можно на рисунке 2.2.1

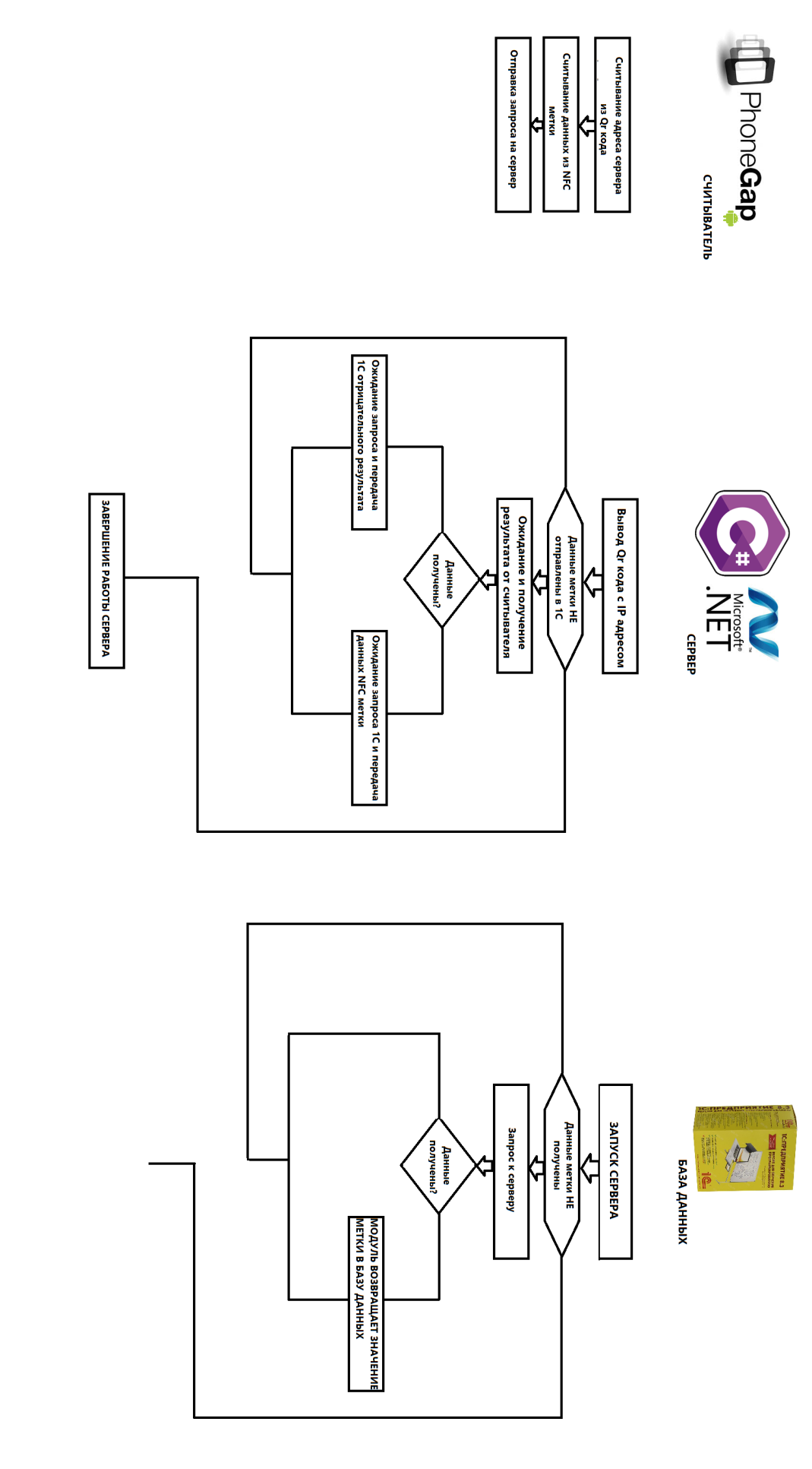


Рисунок 2.2.1. Примерная блок-схема взаимодействия

### 2.2.2 Описание бизнес-логики разрабатываемой конфигурации

Данная информационная система представляет собой конфигурацию для платформы 1С:Предприятие, программный модуль, устанавливаемый на одну ЭВМ с платформой, приложение на мобильный телефон с NFC считывателем и операционной системой Android, устанавливаемое отдельно. Межмодульное взаимодействие осуществляется средствами IP версии протокола 4 по локальной сети. В связи с тем, что часть модулей развертывается на мобильном устройстве, для организации локальной сети подходит только сеть Wi-Fi.

Конфигурация устанавливается штатными средствами платформы 1С:Предприятие – через конфигуратор. Программный модуль “сервер” устанавливается администратором, более подробно о этом написано в соответствующем разделе.

Мобильное приложение ставится через любой файловый менеджер или средство отладки ADB, предварительно необходимо разрешить установку из неизвестных источников. В процессе использования устройство должно быть непрерывно подключённым к одной Wi-Fi сети с ЭВМ, с работающей конфигурацией платформы 1С и программой-сервером.

Условный охранник, работающий на контрольно-пропускном пункте колледжа, осуществляет учет студентов, осуществляющих вход и выход. Для этого в конфигурацию занесен специальный общий модуль, вызываемый кнопкой на форме. Модуль запускает сервер, дальнейшее взаимодействие схематически обозначено на рисунке 2.1.1. После нажатия на кнопку на экране появляется специальное окно с Qr кодом. Код считывается мобильным приложением, далее к телефону прикладывается условная карта студента, выраженная проездным билетом “Тройка”. После считывания, уникальный идентификатор карты передается в общий модуль платформы 1С:Предприятие, который возвращает результат в событие нажатия на кнопку. Дальнейшая работа определяется логикой обработчика события.

## 2.3. Проектирование программного обеспечения

Данное программное обеспечение разделено на независимые модули. Всего информационная система содержит три модуля: мобильное приложение(считыватель меток), программа с Qr кодом для подключения (сервер) и конфигурация 1С:Предприятие

Начать разбор исходных кодов следует с реализации мобильного приложения, так как его логика наиболее простая. Убедиться в этом можно на рисунке 2.2.1. В качестве средства разработки мобильного приложения был выбран Adobe PhoneGap, так как

1. Не требует установки средств разработчика на машину, где осуществляется сборка конечной программы

Существует бесплатный сервис, который осуществляет сборку приложений прямо в браузере. Он размещен по адресу <http://build.phonegap.com/>. Рассмотреть его скриншот можно на рисунке 2.3.1

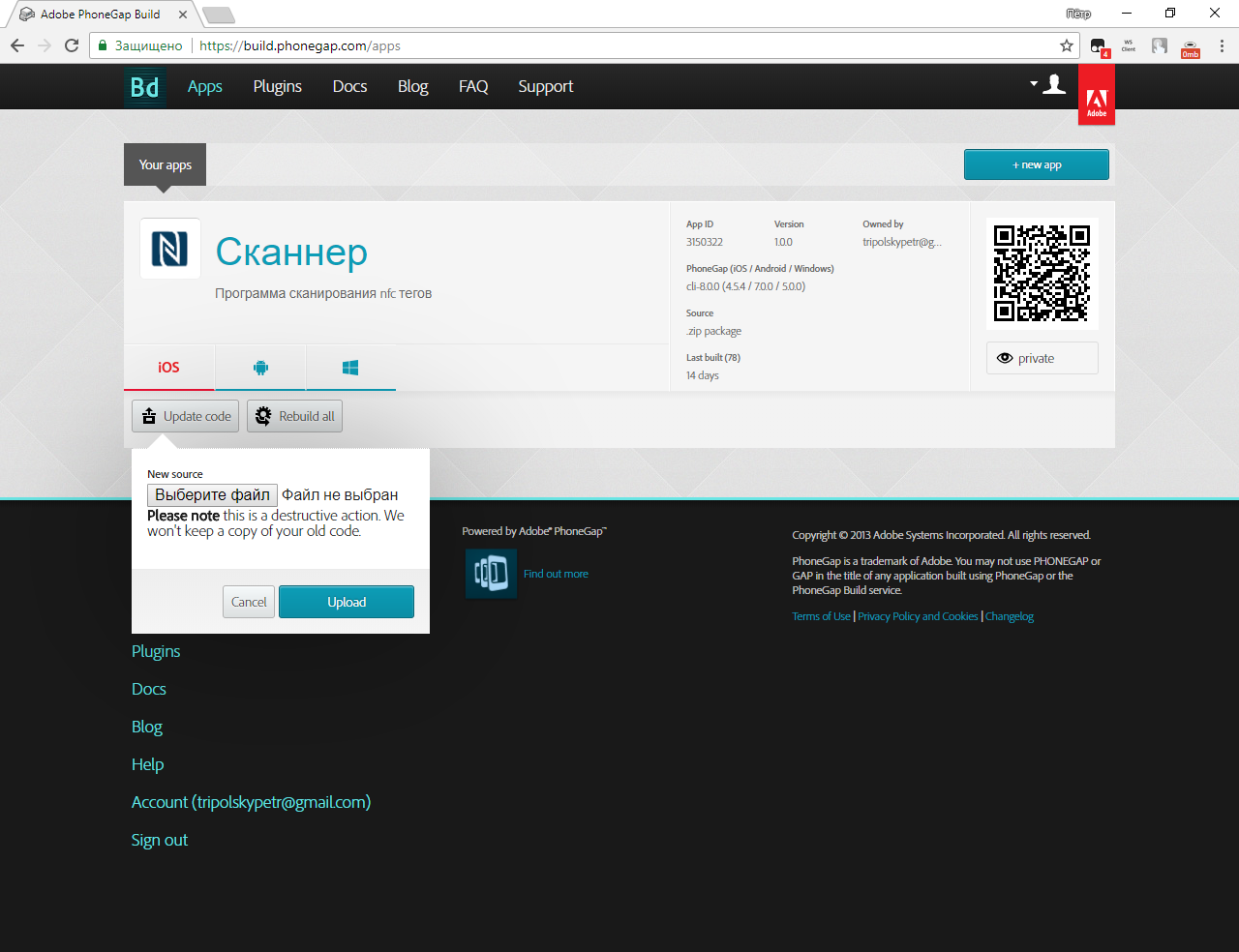


Рисунок 2.3.1. Сборка из исходных кодов онлайн

Исходный код, в данном случае, представляет собой zip архив с файлами html, css, javascript и файлом конфигурации config.xml. Данный язык был выбран не случайно, так как он похож на язык платформы 1С:Предприятие – интерпретатор сам определяет типы переменных, стиль написания кода процедурный.

Файл конфигурации содержит настройки для сборщика приложения. Приложения можно собрать как под Android, так и под iOS и Windows Phone. Более подробно ознакомиться с исходным кодом мобильного считывателя можно на приложенном диске в директории /NfcServer/NfcClient/.

1. Полностью решает поставленные задачи создания пользовательского интерфейса – можно работать с камерой.

Пользовательский интерфейс создается с применением верстки на языке разметки HTML, который легок в освоении и включает в себя большое множество средств оформления текста, что приветствуется для решения задач информационного вывода. Так как мобильное приложение осуществляет всего три действия, простота пользовательского интерфейса очень важна. После правильной настройки управлять мобильным приложением можно не пользуясь сенсорным экраном, лишь приложив NFC метку к считывателю устройства

Реализация сервера будет создана с применением языка программирования C#, так как для этого языка программирования доступны возможности .Net Framework. При программировании на ОС семейства Windows NT очень важно применять родные средства разработки для обеспечения максимальной производительности. Разработка сервера будет вестись в среде разработчика Microsoft Visual Studio Community 2017, так как она позволяет работать с пакетным менеджером NuGET, из которого будет получено средство генерации Qr кодов. Классы для работы с протоколом HTTP находятся в стандартной библиотеке, что так же удобно. Посмотреть скриншот этой среды разработчика можно на рисунке 2.3.2

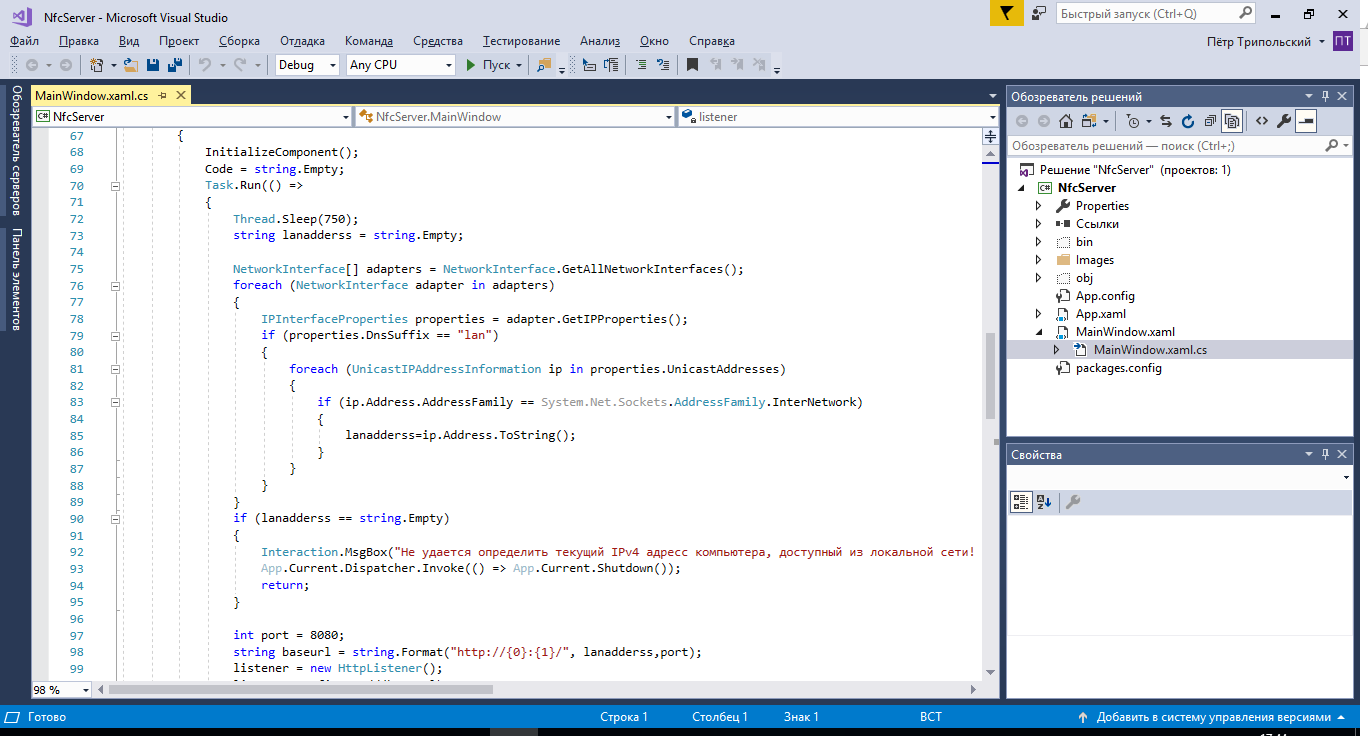
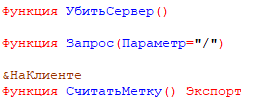


Рисунок 2.3.2. Работа в среде разработчика Microsoft Visual Studio

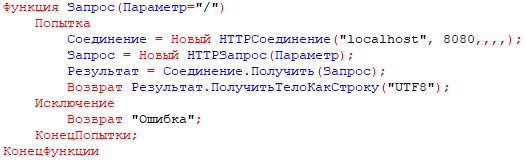
Среда разработчика Microsoft Visual Studio позволяет проводить пошаговую отладку, получать всю информацию о исключениях, считывать значение переменных в процессе исполнения, создавать дамп памяти программы, вести мониторинг потребляемой памяти в реальном времени и другие операции, помогающие программисту создать отказоустойчивое ПО, которое можно интегрировать в платформу 1С:Предприятие, не снизив общий уровень надежности информационной системы как общности программ в целом.

Конфигурация платформы 1С:Предприятие для работы с этой информационной экосистемой должна содержать специфичный общий модуль, реализующий работы с программой-сервером и предусматривающим множество вариантов развития событий. Модуль содержит в себе три функции:

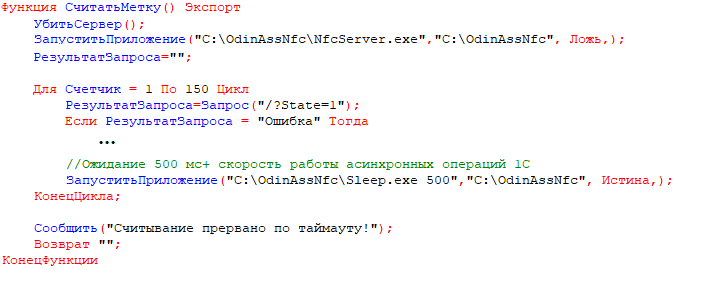


Функция “УбитьСервер” запускается после вызова функции “СчитатьМетку”. Она работает синхронно и завершает все экземпляры сервера, чтобы избежать коллизий.

Функция “Запрос” осуществляет запрос на сервер. В ней есть обработчик исключений – если сервер не ответил, она вернет значение-заглушку “Ошибка”. Это необходимо для логического ветвления в функции “СчитатьМетку”. Её реализация представлена ниже:



Реализация функции “СчитатьМетку” длинная, поэтому я разберу лишь частично. Функция запускается в контексте клиента, чтобы она могла выводить на экран сообщения помимо возвращаемого значения. Если произошла ошибка, то функция вернет пустую строку. Внутри функции есть не бесконечный цикл. В конце каждой итерации происходит задержка 500 миллисекунд, чтобы не перегружать сервер запросами. Внутри итерации логическим ветвлением перебраны все варианты развития событий. Если удалось получить данные из метки, функция тут же возвращает значение. Примерную реализацию функции можно увидеть ниже:



## 2.4 Разработка пользовательских интерфейсов программного обеспечения

В данной информационной системе большое множество пользовательских окон. Пользователь взаимодействует с ними с использованием манипулятора управления типа “Мышь” или сенсорным экраном. Скриншоты пользовательских интерфейсов можно посмотреть на рисунках 2.4.1

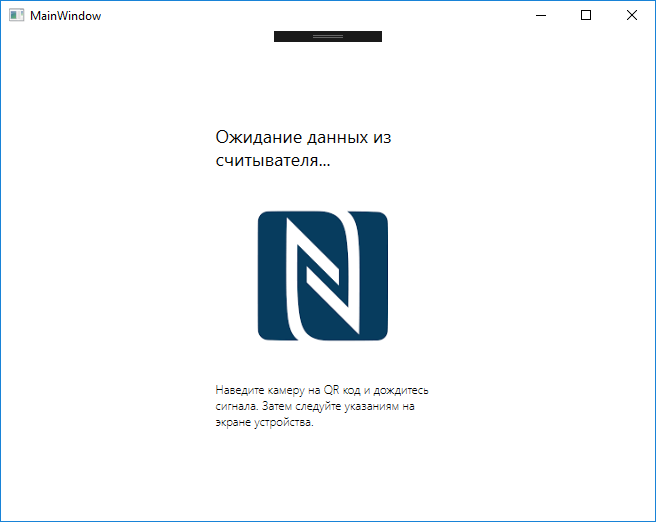


Рисунок 2.4.1. Программа “Сервер” до определения IP адреса.

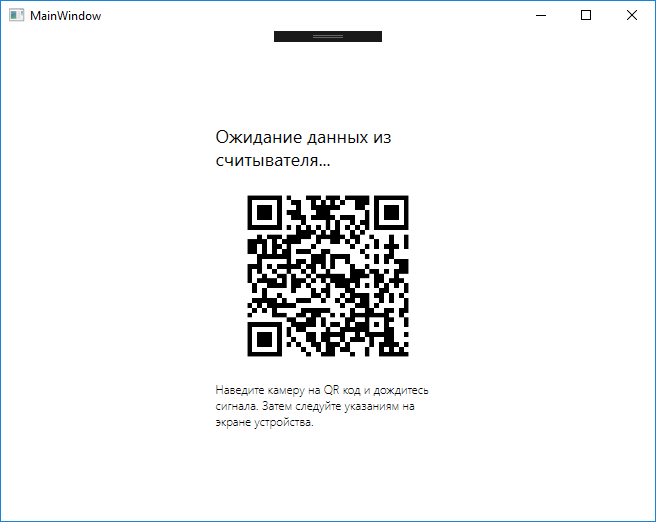


Рисунок 2.4.2. Программа “Сервер” после определения ip адреса

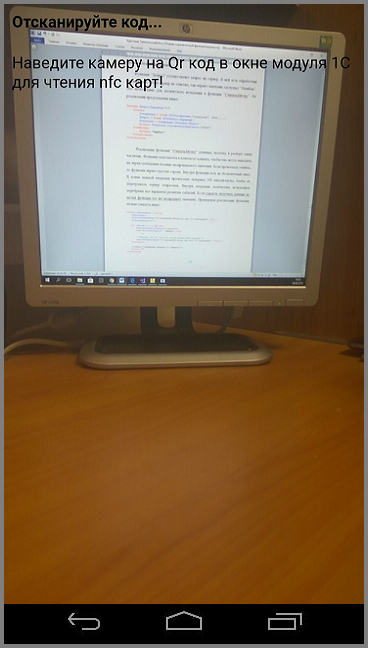


Рисунок 2.4.3. Мобильное приложение “Считыватель меток”, направленное на документ Microsoft Office Word на компьютере.



Рисунок 2.4.4. Мобильное приложение “Считыватель меток” в ожидании сканирования NFC тега.

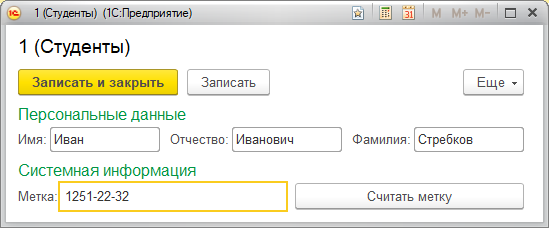


Рисунок 2.4.5. Форма программы на платформе 1С:Предприятие с интегрированным модулем с считывателем меток

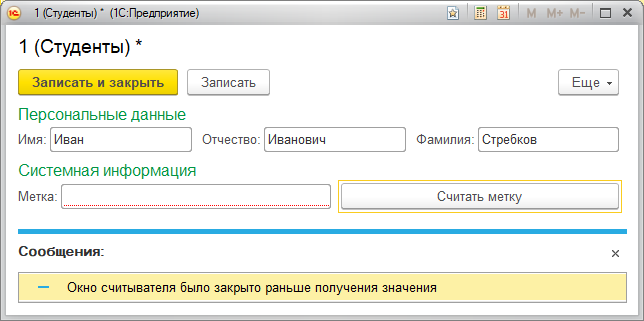


Рисунок 2.4.6. Окно программы-сервера было закрыто до получения значения.

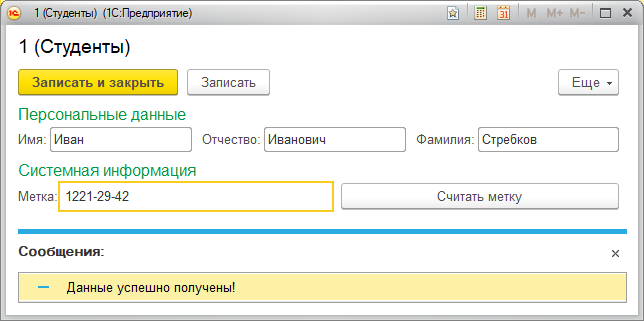


Рисунок 2.4.7. Данные успешно получены.

## 2.5. Организация защиты базы данных

Платформа 1С:Предприятие обеспечивает должный уровень отказоустойчивости и безопасности, проверенный годами работы в отечественных системах учета товаров на складах, бухгалтерии и прочем ПО, работающим с базами данных. Администратор может увеличить скорость работы платформы, подключив её к реляционной базе данных. При этом уровень безопасности только увеличивается.

## 2.6. Тестирование и отладка программного обеспечения

С результатами тестирования данного программного обеспечения можно ознакомиться на рисунке 2.6.1. Данная информационная система обеспечивает высокий уровень отказоустойчивости.



Рисунок 2.6.1. Результаты тестирования

# 3. РУКОВОДСТВО ПО ИСПОЛЬЗОВАНИЮ ПРОГРАММЫ

## 3.1. Руководство администратора

### 3.1.1 Общие сведения о программе

Данная информационная система состоит из трех частей: конфигурации платформы 1С:Предприятие, мобильного приложения на Android для считывания NFC карт и HTTP сервера для их взаимодействия. Они взаимодействуют по локальной сети. Так как мобильные устройства обычно не обладают разъемом RJ45, предполагается, что локальная сеть организована по Wi-Fi. При разработке использовались кроссплатформенные технологии, но финальная дистрибуция выпущена только под Windows 10 Pro версии 1709. Программа для считывания карт устанавливается как и другие и специфичной настройки не требует.

### 3.1.2 Настройка Firewall

Чтобы программные модули могли взаимодействовать друг с другом, требуется специфично настроить операционную систему. Для начала, нужно задать адаптеру локальной сети специальный суффикс “lan”. Это нужно для того, чтобы программа “сервер” могла отличить реальный адаптер локальной сети от эмулируемого через VirtualBox. Текущий суффикс можно узнать консольной командой ipconfig (рисунок 3.1.1)

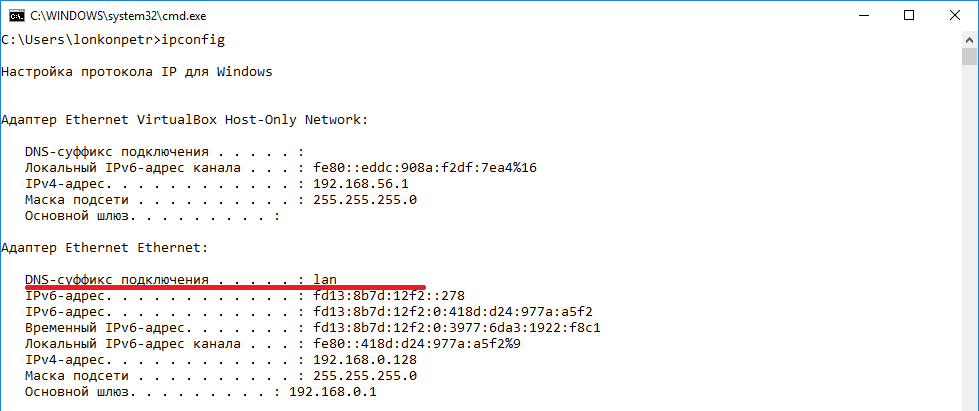


Рисунок 3.1.1. Правильно настроенный dns суффикс.

Настроить его можно в программе “Панель управления” по адресу “Панель управления\Сеть и Интернет\Сетевые подключения”. Далее необходимо выбрать правой кнопкой мыши интересующий адаптер сети интернет, перейти в “Свойства”, там выбрать компонент “IP версии 4(TCP/IPV4)”, нажать на “Свойства”, в новом диалоговом окне нажать на “Дополнительно”, перейти на вкладку DNS и внизу в поле ввода “DNS-суффикс подключения” ввести “lan”. После подтвердить открытые ранее диалоговые окна кнопкой “ОК”. Для наглядности используйте схему на рисунке 3.1.2

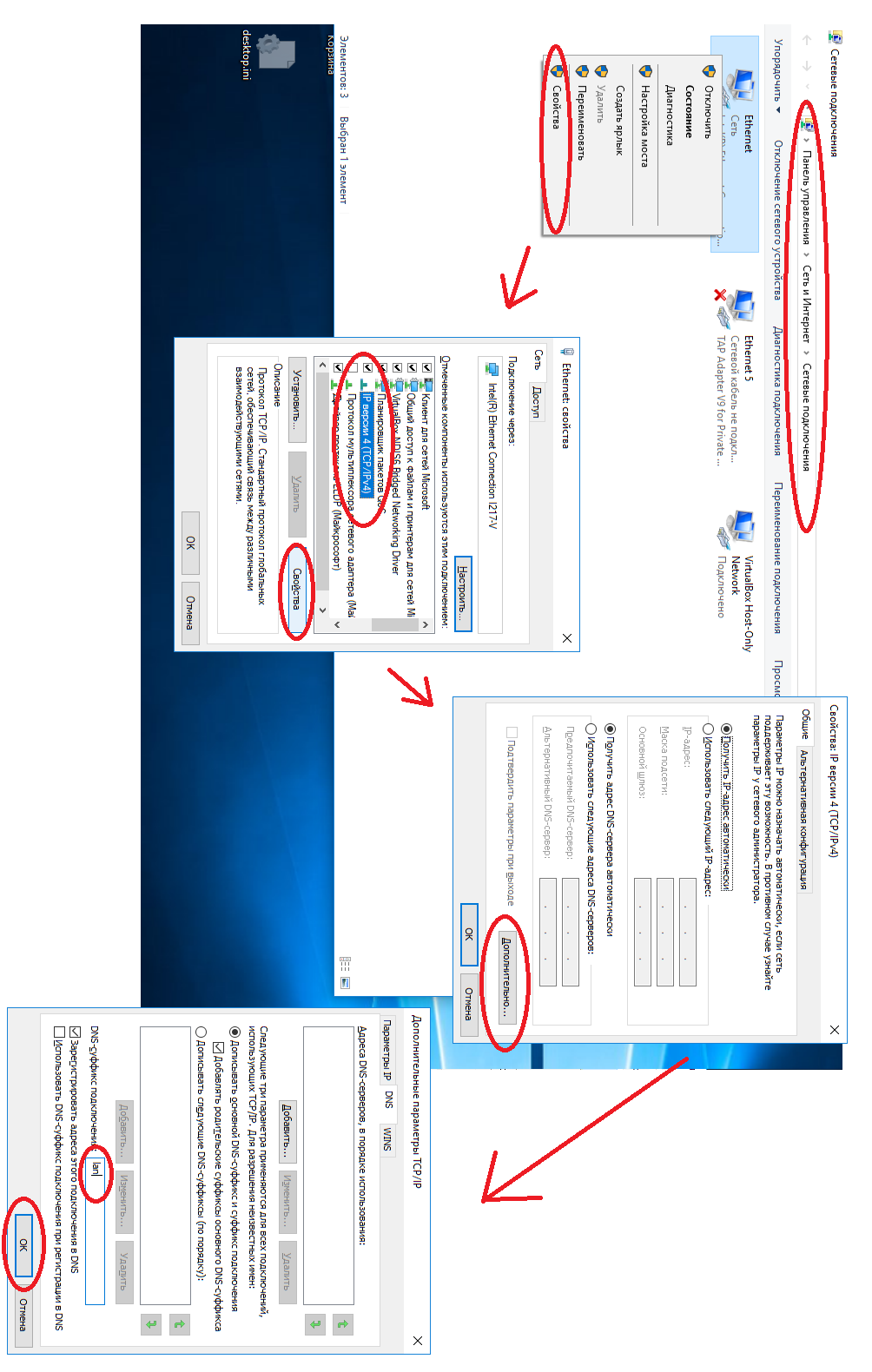


Рисунок 3.1.2. Настройка DNS суффикса

Далее необходимо открыть на компьютере порт 8080. Для этого откройте программу “Монитор брандмауэра Защитника Windows в режиме повышенной безопасности”. Нажмите правой кнопкой мышки на пункт меню слева “Правило для входящих соединений”, далее “Создать правило”. Выберите в списке “Для порта”, нажмите “Далее”. Выберите “Протокол TCP”, введите число 8080 в текстовое поле, нажмите “Далее”. Выберите “Разрешить подключение” и нажмите “Далее”. Убедитесь, что все чекбоксы выбраны и нажмите “Далее”. Введите произвольное название и завершите настройку кнопкой “Готово”. Повторите действия для пункта меню “Исходящие соединения”. Подробнее можно увидеть на рисунке 3.1.3

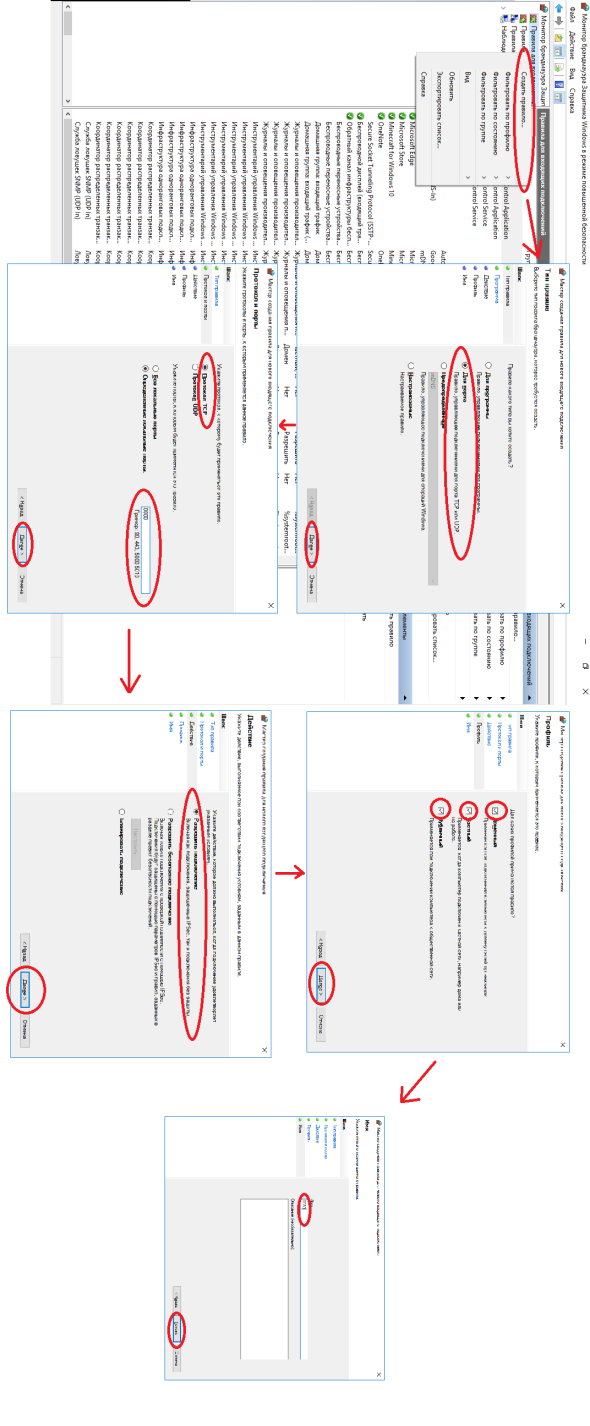


Рисунок 3.1.3. Настройка портов.

### 3.1.3 Инструкция по развертыванию информационной базы 1С

Конфигурация базы данных 1С:Предприятие настраивается как и любые другие. Открывается конфигуратор, в контекстном мену выбирается пункт “Администрирование”. Из выпадающего меню выбирается “Загрузить информационную базу”. Эти действия изображены на рисунке 3.1.4

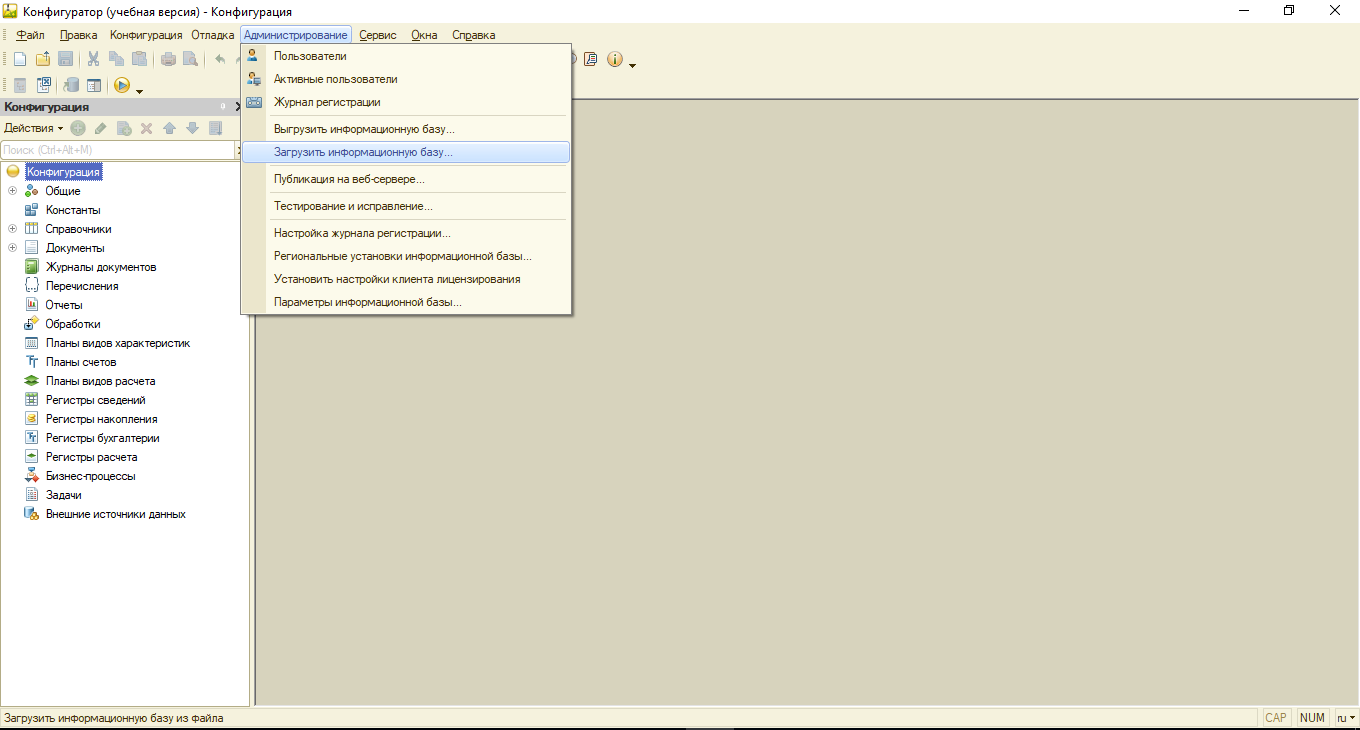


Рисунок 3.1.4. Установка готовой конфикурации.

После установки конфигурации необходимо так же распаковать содержимое zip архива Программа.zip с диска поставки из директории NfcServer/NfcServer на текущий компьютер по данному адресу: “C:\OdinAssNfc”. Чтобы создать директорию на диске C: потребуются права администратора!

Первый запуск программы должен производить администратор, непосредственно кликнув правой кнопкой мыши по исполняемому файлу NfcServer.exe и выбрав “Запустить от имени администратора”. Программа зарезервирует в реестре обработчики адресов для запуска веб-сервера, запустив консольную команду вида “cmd /c netsh http add urlacl url=http://\*:8080/user=ИмяПользователя”. Процедуру необходимо выполнять каждый раз, когда ip адрес ЭВМ меняется. Поэтому сервер или должен не отключаться от сети WiFi или ему требуется задать статичный ip4 адрес. Делается это в том же диалоговом окне, что и DNS суффикс, без перехода в поддиалог “Дополнительно”.

### 3.1.4 Интеграция модуля считывания NFC карт в новые конфигурации

Для интеграции данного модуля в первую очередь нужно правильно настроить текущую конфигурацию и проверить её работоспособность. Данная конфигурация является тестовой средой для данного модуля. После чего, необходимо из текущей конфигурации из общего модуля “СчитывательМеток” скопировать весь код в новый общий модуль конфигурации, где данное множество программ необходимо интегрировать. Посмотреть расположение общего модуля можно на рисунке 3.1.5.

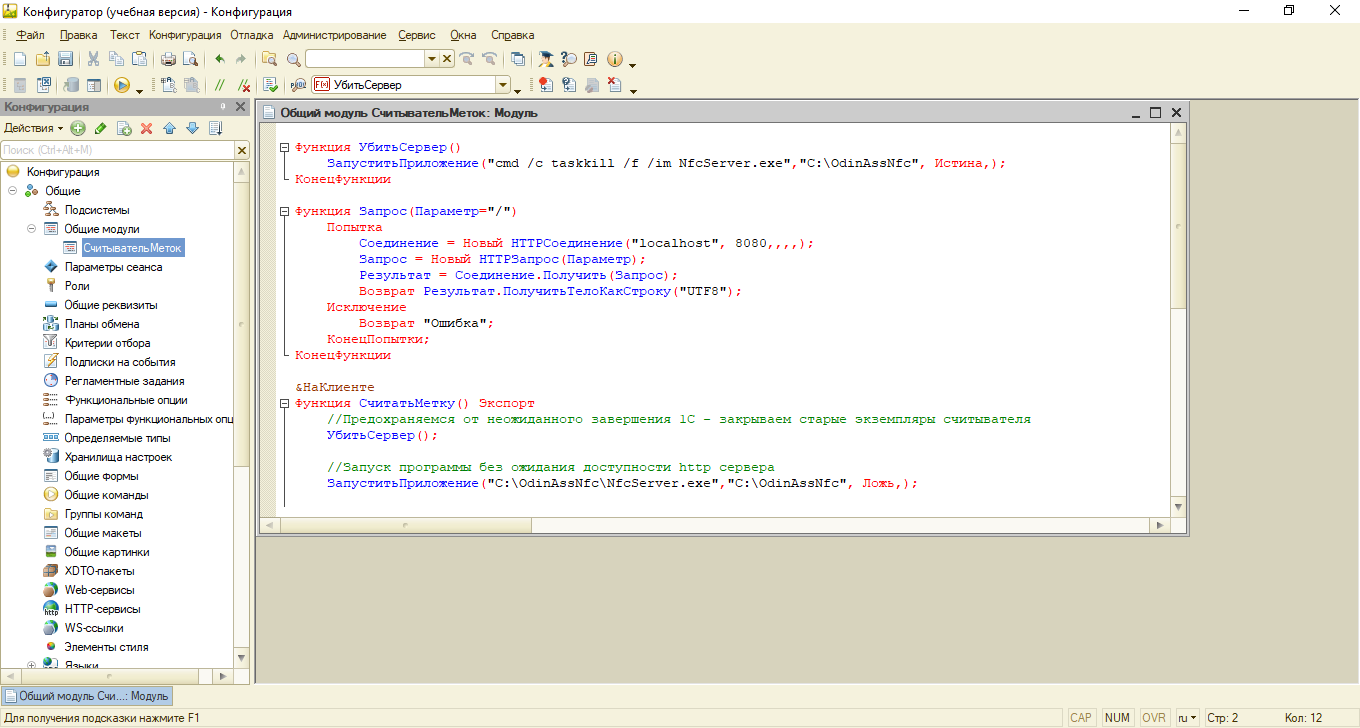


Рисунок 3.1.5. Расположение общего модуля

Далее необходимо в новой конфигурации включить “Режим использования синхронных вызовов расширений платформы и внешних компонент”. Делается это в свойстве конфигурации, в самом низу. Подробнее можно увидеть на рисунке 3.1.6.

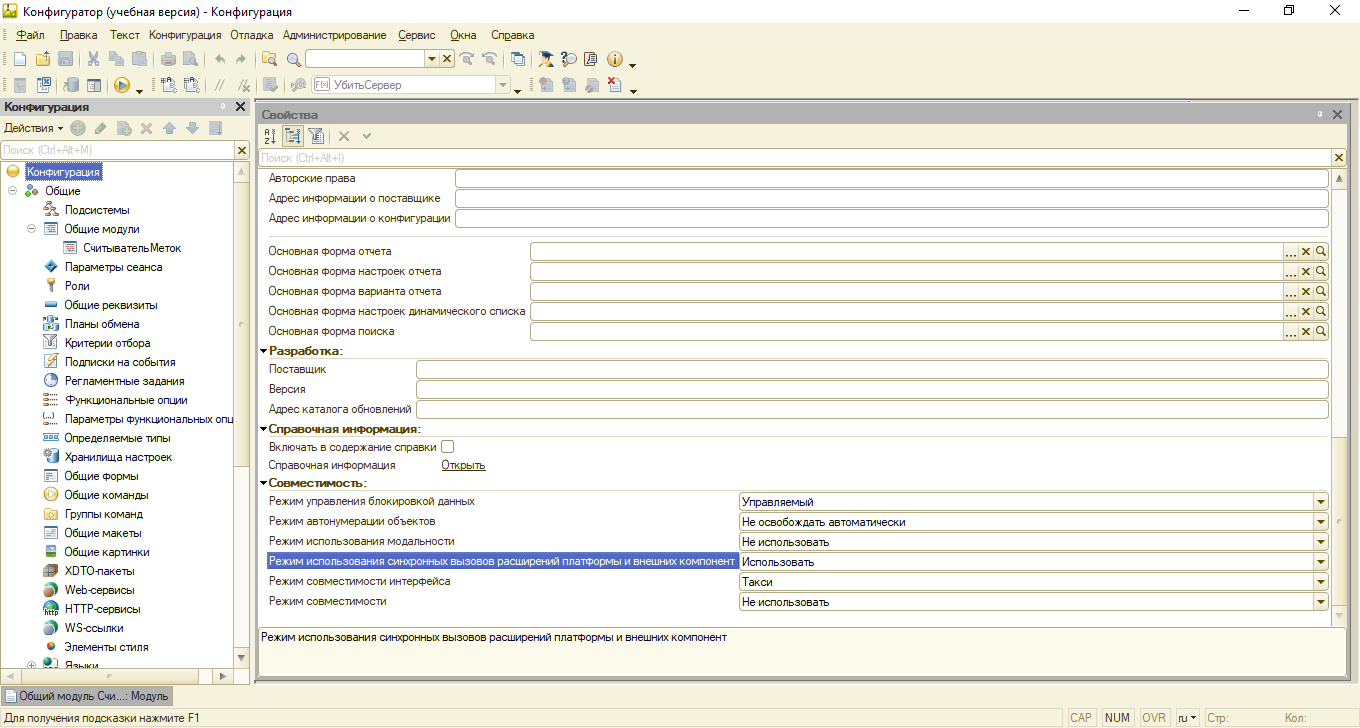


Рисунок 3.1.6. Режим использования синхронных вызовов расширений платформы и внешних компонент

## 3.2. Руководство пользователя

### 3.2.1 Считывание номера метки

В процессе работы с данной информационной системой потребуется считывать данные из NFC метод. Для этого необходимо нажать на кнопку “Считать метку”, находящуюся в диалоговом окне, навести камеру из мобильного приложения “Сканер меток” на появившийся в диалоговом окне Qr код и приложить метку к устройству. Более подробно увидеть работу можно на рисунке 3.2.1

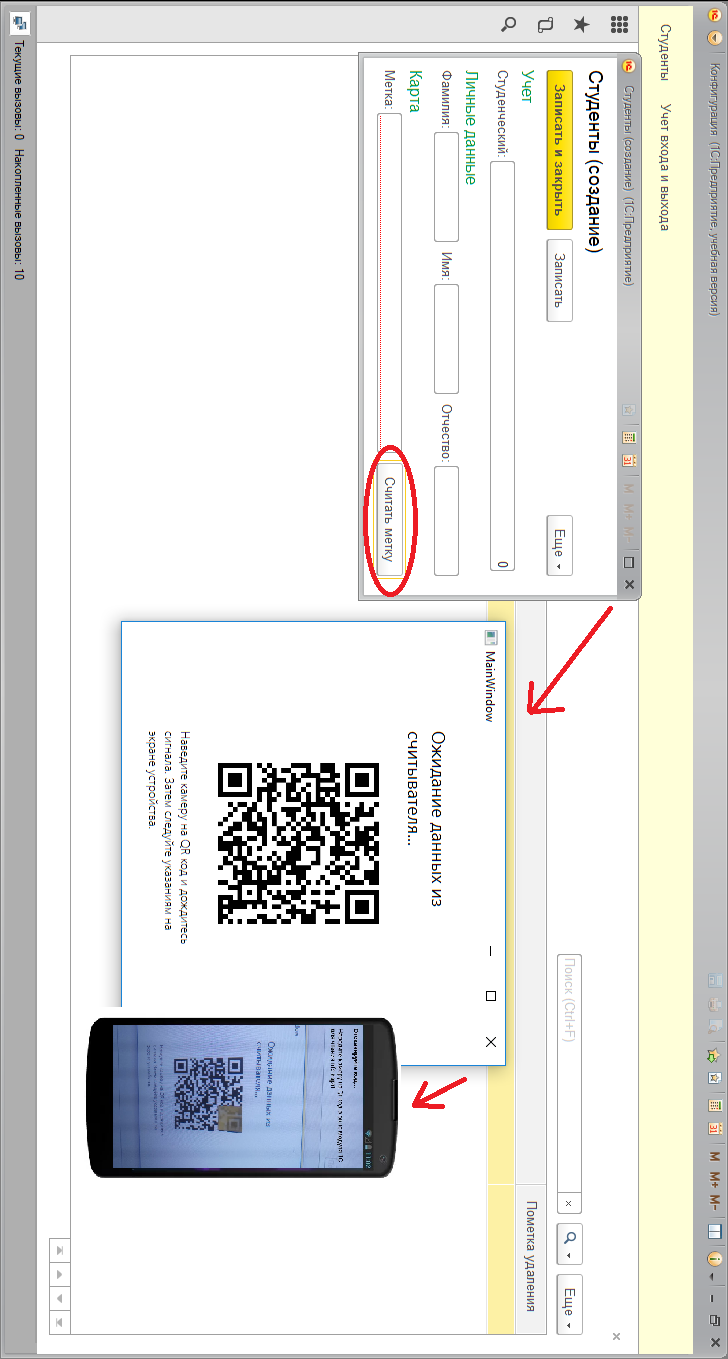


Рисунок 3.2.1. Cчитывание метки

### 3.2.2 Ведение учета

Сначала требуется ввести в информационную систему список студентов. Сделать это можно в справочнике “Студенты”, подробнее ознакомиться с порядком действий можно на рисунке 3.2.2.

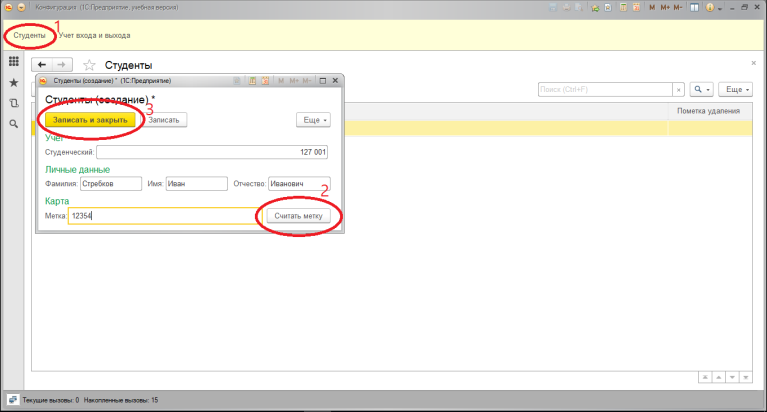


Рисунок 3.2.2. Ввод данных о студентах

После программу необходимо передать в эксплуатацию охране на контрольно-пропускном пункте колледжа. Они, в свою очередь, будут работать с вкладкой “Учет входа и выхода”. Охрана должна впускать или выпускать студента только после учета в базе данных. Идентифицировать студента можно как по номеру студенческого билета, так и NFC карте. Посмотреть скриншот формы можно на рисунке 3.2.3.

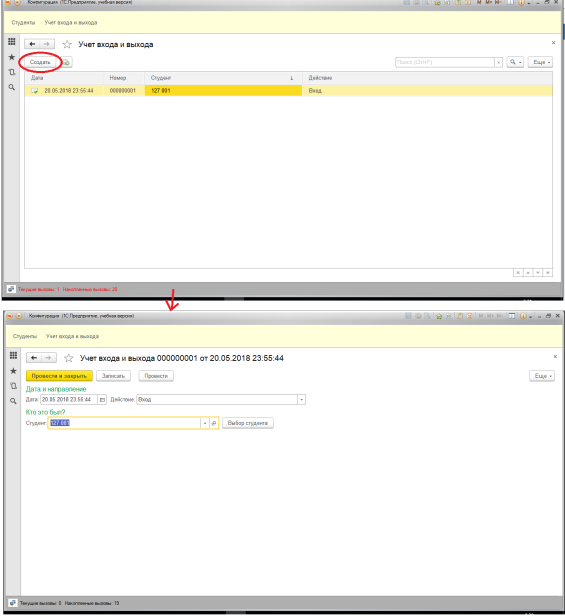


Рисунок 3.2.3. Учет студентов.

# ЗАКЛЮЧЕНИЕ

Во время выполнения курсовой работы были достигнуты поставленные цели и задачи. Была создана программа на платформе 1С:Предприятие. Благодаря этой программе условные охранники, работающие на КПП колледжа, смогут вести учет студентов, как по интерактивным картам, так и номерам из студенческих билетов

При разработке базы данных были изучены интернет статьи о работе с интеграцией платформы 1С:Предприятие с другими программными средствами.

Данная информационная система:

* Отказоустойчива
* Проста в использовании.

Достоинства модуля:

* Универсальный, работает на ОС семейства Windows NT;
* Совместим с любыми другими конфигурациями;
* Работает в Учебной Версии 1С:Предприятие.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

**Стандарты**

1. ГОСТ 7.1. – 2003. Библиографическая запись. Библиографическое описание. Общие требования и правила составления. – М.: ИПК Издательство стандартов, 2004. – 169 с.
2. ГОСТ 7.32 – 2001. Система стандартов по информацию, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления. – М.: ИПК Издательство стандартов, 2001. – 21 с.
3. ГОСТ 7.82 – 2001. Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления. – М.: ИПК Издательство стандартов, 2001. –21 с.
4. Единая система программной документации. – М.: Стандартинформ, 2005. –128 с.

**Монографии, учебники, учебные пособия**

1. М.Г. Радченко, Е.Ю. Хрусталева. 1C:Предприятие 8.3. Практическое пособие разработчика. Примеры и типовые приемы. Издательство: 1С-Паблишинг , 2013, с.965; Е.Ю.
2. Е.Ю. Хрусталева. Язык запросов «1С:Предприятия 8». Издательство: 1С-Паблишинг, 2013, с.369;
3. ООО «1С-Учебный центр №3». Введение в конфигурирование в системе "1C:Предприятие 8". Основные объекты. Издательство: 1С-Паблишинг, 2014, с.177;

# ПРИЛОЖЕНИЕ А

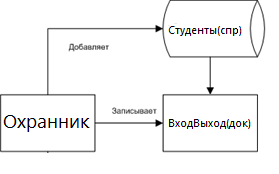


Рисунок А.1 - Концептуальная схема