Частное учреждение образования

«Колледж бизнеса и права»

РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ «RESTFUL API СЕРВИСА

ДЛЯ ОТСЛЕЖИВАНИЯ ПЕРЕМЕЩЕНИЯ ЖИВОТНЫХ»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по дисциплине

«Конструирование программ и языки программирования»

КП.091005.401

Руководитель проекта (Е.Н.Коропа)

Учащийся (А.О.Зеневич)

2023

Содержание

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*2*

*КП Т.091005.401*

*Разраб.*

*Зеневич А.О.*

*Провер.*

*Коропа Е.Н.*

*Реценз.*

*Н. Контр.*

*Утверд.*

Разработка web-приложения «restful api сервиса для

отслеживания перемещения животных»

*Лит.*

*Листов*

*КБП*

*22*

*у*

X

# Введение

В настоящее время развитие информационных технологий оказывает существенное влияние на различные сферы жизни человека, в том числе и на животноводство. Современные технологии позволяют создавать программные средства, способные оптимизировать процессы управления и контроля за животным миром, а также предоставлять эффективные инструменты для отслеживания перемещения и жизненных циклов животных.

RESTful API Service — это тип веб-сервиса, который использует архитектурный стиль REST (Representational State Transfer), чтобы обмениваться данными между клиентом и сервером. RESTful API Service предоставляет универсальный набор правил для создания и обмена информацией через сеть, используя HTTP протокол.

Основные принципы RESTful API Service включают использование URI (Uniform Resource Identifier) для идентификации ресурсов, использование HTTP методов (GET, POST, PUT, DELETE) для выполнения операций с ресурсами, а также использование форматов данных, таких как JSON или XML, для обмена информацией.

RESTful API Service обеспечивает более простой и легковесный подход к созданию веб-сервисов, что позволяет разработчикам быстрее создавать и внедрять приложения. Этот подход также облегчает интеграцию между различными приложениями и платформами, так как RESTful API Service является независимым от языка и платформы решением для обмена данными.

В рамках данного курсового проекта ставится задача разработки RESTful API сервиса, который будет предоставлять возможность просмотра информации о животных, их типах, местоположении и перемещениях. Реализуемый сервис позволит осуществлять поиск, создание, изменение и удаление данных о животных, их типах, точках местоположения и местах посещения животных.

Актуальность поставленной задачи обусловлена необходимостью создания эффективных инструментов для управления и контроля за животным миром, а также необходимостью автоматизации процессов сбора и обработки информации, связанной с животными. Кроме того, разработка подобных сервисов способствует развитию индустрии информационных технологий и расширению возможностей их применения в различных сферах деятельности.

К курсовому проекту также прилагается пояснительная записка, состоящая из четырех разделов и графической части.

В первом разделе в «Описании задачи» приводится описание предметной области, включающий в себя описание бизнес-процессов, происходящих в области управления и контроля за животным миром, описание алгоритмов получения промежуточных и итоговых показателей, а также обоснование необходимости компьютерной обработки информации или процессов. Определяется потенциальная аудитория, которой будет предоставлен доступ к сервису.

Во второй главе «Проектирование веб-приложения» проводится постановка задачи, включающая в себя определение функций, которые должны быть автоматизированы на основе проанализированных бизнес-процессов и бизнес-задач, а также описание существующих аналогов с указанием отличий, которые будут реализованы в разрабатываемом программном средстве. Перечисляются требования к приложению, описывается проектирование макета веб-приложения, перечисляются программные-технические средства необходимые для разработки приложения. Также приводится способы защиты и сохранности данных и описывается организация и ведение информационной базы приложения.

В третьей главе «Реализация веб приложения» приводится описание разделов веб-приложения, описание навигационного меню, описываются используемые функции и процедуры, а также приводится функциональное тестирование на основании тест-кейсов и скриншотов.

В четвёртом разделе «Применение» описывается назначение веб-приложения, цели разработки, саму установку веб-приложения с описанием всего дерева файлов, описывается область применения, а также класс решаемых задач. Приводятся ограничения, накладываемые на область применения. Обосновываются требования к необходимому для веб-приложения программно-аппаратному обеспечению сервера и клиента для корректной работы приложения.

# Описание задачи

## Анализ предметной области

В разрабатываемом web-приложении точки текущего местоположения животных будут заданы: импортироваться из БД.

В данном разделе рассмотрим предметную область проекта, то есть перемещение животных, а также бизнес-процессы, происходящие в этой области.

Жизненный цикл животных является важным аспектом в их уходе и воспроизведении. Он включает в себя такие этапы, как рождение, рост, взросление, размножение, старение и смерть. В каждом этапе жизненного цикла животных существуют определенные потребности и особенности, которые нужно учитывать в процессе ухода за ними.

Перемещение животных также является важным аспектом. Оно может быть вызвано различными причинами, такими как поиск еды, воды или укрытия, а также в связи с процессом размножения. Кроме того, перемещение животных может быть связано с миграцией, изменением климатических условий или просто с природным инстинктом животного.

Для эффективного ухода за животными необходима компьютерная обработка информации о их перемещении и жизненном цикле. Поэтому, в рамках нашего курсового проекта, мы решаем задачу разработки RESTful API сервиса для отслеживания перемещения и жизненных циклов животных.

При проектировании данного сервиса необходимо учитывать потенциальную аудиторию, которая может использовать его в своей работе. В первую очередь это могут быть зоопарки, ветеринарные клиники, рыбхозы, птицефабрики и т.д.

## Постановка задачи

Исходя из исследований предметной области в предыдущем разделе пояснительной записки сформулируем основные функции приложения и …. Функция 1…

Первоочередной функцией сервиса является просмотр информации о животном. Пользователь должен иметь возможность получать информацию о конкретном животном, включая его тип, текущее местоположение, а также все места, которые он посетил.

Далее следует функция поиска, создания, изменения и удаления животного. Пользователь может осуществлять поиск животного по его текущему местоположению, а также по всем местам, которые он посетил. В случае, если животное не найдено, пользователь может создать новую запись о животном. В случае изменения информации о животном, пользователь может обновить информацию о его типе, местоположении или посещенных местах. При необходимости пользователь может удалить запись о животном.

Также необходимо автоматизировать функции создания, изменения и удаления типа животного. Пользователь может просмотреть информацию о типе животного, а также создать новый тип, изменить его параметры или удалить его.

Функционал сервиса включает также точки местоположения, которые представляют собой географические координаты мест, где были замечены животные. Пользователь может просмотреть информацию о точке локации, создать новую точку, изменить ее параметры или удалить ее из системы.

Наконец, сервис должен предоставлять функции по созданию, изменению и удалению места посещения животных. Пользователь может просмотреть информацию о перемещении животного, создать новую запись о посещенном месте, изменить ее параметры или удалить ее.

При проектировании разрабатываемого программного средства необходимо провести анализ существующих аналогов с указанием отличий, которые будут реализованы в разрабатываемом сервисе.

# Проектирование веб-приложения

## Проектирование модели

Требуется раскрыть суть спроектированных диаграмм. Разработанные диаграммы приводятся в графической части на соответствующих листах. После описания делается ссылка на соответствующий лист в графической части.

Отразить следующие диаграммы:

* вариантов использования с указанием типов связей между вариантами использования. Для основных прецедентов по тексту пояснительной записки необходимо составить сценарии;
* классов с выделением необходимых атрибутов и/или методов, иллюстрирующую модель данных;

структура веб-приложения

## Требования к веб-приложению

Требования к стилистическому оформлению:

* + веб-приложение должно иметь современный, привлекательный и профессиональный дизайн.
  + использование цветовой палитры, соответствующей тематике приложения и создающей приятную атмосферу.
  + согласованный стиль оформления элементов интерфейса для обеспечения удобства использования.

Требования к графическому дизайну:

* + использование графических элементов, подходящих для темы животных и их перемещения.
  + четкое и качественное отображение графики на различных устройствах и экранах.

Требования к шрифтовому оформлению:

* + использование читаемых и понятных шрифтов с хорошей читабельностью.
  + размеры шрифтов должны быть оптимальными для удобного чтения контента.

Требования к средствам просмотра:

* + веб-приложение должно корректно отображаться и работать на различных веб-браузерах (например, Google Chrome, Mozilla Firefox, Microsoft Edge).
  + приложение должно быть адаптивным, чтобы хорошо отображаться на различных устройствах, включая мобильные телефоны и планшеты.

Требования к контенту (наполнению):

* + предоставление информации о животных и их перемещении.
  + понятная и структурированная презентация контента для удобного восприятия и понимания пользователем.

Требования к системе управления контентом:

* + интуитивно понятный интерфейс управления контентом для облегчения работы с данными.

Требования к компоновке страниц:

* + логичное расположение элементов интерфейса на страницах приложения.
  + правильное использование разметки HTML и CSS для структурирования и стилизации контента.
  + согласованная и удобная навигация между страницами приложения.

## Структура веб-приложения

Для наглядного представления логической структуры веб-приложения можно использовать многоуровневый список или ментальную карту. В случае RESTful API сервиса для отслеживания перемещения животных, логическая структура может быть представлена следующим образом:

1. Уровень маршрутов (Routes):
   * /api/animals: Обработка запросов, связанных с животными.
   * /api/animaltypes: Обработка запросов, связанных с типами животных.
   * /api/locations: Обработка запросов, связанных с перемещением и местоположением животных.
   * /api/movementpoints: Обработка запросов, связанных с точками перемещения животных.
2. Уровень контроллеров (Controllers):
   * AnimalsController: Обработка запросов, связанных с животными.
   * AnimalTypesController: Обработка запросов, связанных с типами животных.
   * LocationsController: Обработка запросов, связанных с перемещением и местоположением животных.
   * MovementPoints: Обработка запросов, связанных с точками перемещения животных.
3. Уровень моделей (Models):
   * Animal: Модель, представляющая информацию о животном.
   * AnimalTypes: Модель, представляющая информацию о типах животных.
   * Location: Модель, представляющая информацию о перемещении и местоположении животного.
   * MovementPoint: Модель, представляющая информацию о точках перемещения животных.
4. Уровень хранилища данных (Data Storage):
   * База данных или другое хранилище, где хранится информация о животных, перемещении и жизненных циклах.

Структурная схема приложения в виде диаграммы flowchart с указанием взаимосвязи структурных блоков:

## Проектирование макета веб-приложения

x

## Программно-технические средства, необходимые для разработки приложения

Ниже описаны инструменты разработки, которые будут использоваться для написания программного средства.

Персональный компьютер со следующей аппаратной конфигурацией:

* процессор Intel Core i7-4700HQ 2400 МГц, 2013 года выпуска, техпроцесс 22 нанометра, частота 3.4 ГГц, 4 ядра, 8 потоков;
* оперативная память DDR3L 8 ГБ, тактовая частота 1600 МГц;
* видеокарта Nvidia GeForce GTX 850M 4 ГБ;
* винчестер SSD 480 ГБ.

Программное обеспечение:

* операционная система Windows 10;
* среда программирования Visual Studio 2022 c использованием интерфейса программирования приложений API Windows Forms;
* язык программирования C#;
* система управления базами данных SQLite 3;
* интегрированная среда SQLite Studio;
* программная платформа Microsoft .NET Framework 6.0;
* офисный пакет приложений Microsoft Office (MS Office) 2022;
* программа для построения диаграмм Drawio.io.

Операционная система – это набор управляющих программ, предназначенных для управления ресурсами вычислительной системы как единого комплекса, другими словами, операционная система – это набор программного обеспечения, который обеспечивает работу компьютера.

В ходе выполнения проекта были задействованы следующие подключаемые модули приложения:

1. «Microsoft Visual Studio» ­– это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений, это также линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов.
2. «Guna Framework» – дает вам доступ к неограниченным обновлениям. Оптимизированная производительность. Легкий в Visual Studio и вывод оптимизированных графических интерфейсов. Дружественный и простой для разработчиков.
3. «SQLite» – это быстрая и легкая встраиваемая однофайловая СУБД на языке C, которая не имеет сервера и позволяет хранить всю базу локально на одном устройстве. Для работы SQLite не нужны сторонние библиотеки или службы. Понятие «встраиваемый» означает, что СУБД не использует парадигму клиент-сервер.
4. «SQLite Studio» – для работы с базой данной (БД) удобно использовать менеджер БД, с помощью которого можно управлять базой в визуальном режиме. Для СУБД SQLite самый удобный менеджер – это SQLite Studio.
5. «System.Windows.Forms» ­– содержит классы для создания приложений на основе Windows, которые в полной мере используют возможности богатого пользовательского интерфейса, доступные в операционной системе Microsoft Windows.
6. «System» ­– интерфейс программирования приложений, отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework. Содержит основные классы и базовые классы, которые определяют часто используемые типы значений и ссылок, события и обработчики событий, интерфейсы, атрибуты и обработку исключений.

«.NET Framework» – [программная платформа](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B0#%D0%BA%D1%80%D0%BE%D1%81%D1%81%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), выпущенная компанией [Microsoft](https://ru.wikipedia.org/wiki/Microsoft) в [2002 году](https://ru.wikipedia.org/wiki/2002_%D0%B3%D0%BE%D0%B4). Основой платформы является общеязыковая среда исполнения [Common Language Runtime (CLR)](https://ru.wikipedia.org/wiki/Common_Language_Runtime), которая подходит для разных языков программирования. Функциональные возможности CLR доступны в любых языках программирования, использующих эту среду.

Чаще всего это четыре языка из семейства .NET:

* C#;
* F#;
* Visual C++;
* Visual Basic.

«Microsoft Office 2022» – офисный пакет приложений, созданных корпорацией Microsoft для операционных систем Microsoft Windows.

«Draw.io» - используется для создания таких типов диаграмм, как блок-схемы, Организационные диаграммы, Поэтажные планы, Сетевые диаграммы, Диаграммы UML, Интеллектуальные карты и многое другое. Он также обычно используется для таких сценариев, как Отображение процессов и Визуальная совместная работа.

База данных работает под управлением SQLite 3. Используется многопоточный доступ к базе данных. Необходимо обеспечить одновременную работу с программой с той же базой, данной модулей экспорта внешних данных.

Язык для программы используется C#.

Поддерживаемые операционные системы:

* Windows 10 версии 1703 и выше: Домашняя, Pro, для образовательных

учреждений и Корпоративная);

* Windows Server 2016/2019: Standard и Datacenter;
* Windows 8.1 (с обновлением 2919355): Core, Профессиональная и Корпоративная;
* Windows Server 2012 R2 (с обновлением 2919355): Essentials, Standard, Datacenter;
* Windows 7 с пакетом обновления 1 (SP1) (с последними обновлениями Windows): Домашняя расширенная, Профессиональная, Корпоративная, Максимальная.

Оборудование:

* процессор с тактовой частотой не ниже 1,8 ГГц. Рекомендуется использовать как минимум двухъядерный процессор;
* 1 ГБ ОЗУ; рекомендуется 8 ГБ ОЗУ (минимум 2,5 ГБ при выполнении на виртуальной машине);
* требуется 1 ГБ доступного пространства на жестком диске;
* скорость жесткого диска: для повышения производительности установите Windows и Visual Studio на твердотельный накопитель (SSD);
* видеоадаптер с минимальным разрешением 720p (1280 на 720 пикселей); для оптимальной работы рекомендуется разрешение WXGA (1366 на 768 пикселей) или более высокое.

Поддерживаемые языки:

* русский.

## Защита и сохранность данных

x

## Организация и ведение информационной базы (модели)

В таблицах 1-4 представлена организация данных в базе данных «Microsoft SQL SERVER».

Таблица 1 – Сущность «Типы животных»

|  |  |  |  |
| --- | --- | --- | --- |
| Типы животных | | | |
| Название | Тип данных | Ограничение | Ключ |
| ID типа  животного | INT | n N:(n > 0) | Первичный |
| Имя типа  животного | NVARCHAR(50) | NOT NULL |  |
| Описание типа  животного | NVARCHAR(50) | NOT NULL |  |

Таблица 2 – Сущность «Животные»

|  |  |  |  |
| --- | --- | --- | --- |
| Животные | | | |
| Название | Тип данных | Ограничение | Ключ |
| ID животного | INT | n N:(n > 0) | Первичный |
| ID типа  животного | INT | n N:(n > 0) | Внешний |
| Имя животного | NVARCHAR(50) | NOT NULL |  |
| Описание  животного | NVARCHAR(255) | NOT NULL |  |

Таблица 3 – Сущность «Локации»

|  |  |  |  |
| --- | --- | --- | --- |
| Локации | | | |
| Название | Тип данных | Ограничение | Ключ |
| ID локации | INT | n N:(n > 0) | Первичный |
| Имя локации | NVARCHAR(50) | NOT NULL |  |
| Описание  локации | NVARCHAR(255) | NOT NULL |  |
| Широта | DECIMAL(9, 6) |  |  |
| Долгота | DECIMAL(9, 6) |  |  |

Таблица 4 – Сущность «Точки перемещения животного»

|  |  |  |  |
| --- | --- | --- | --- |
| Точки перемещения животного | | | |
| Название | Тип данных | Ограничение | Ключ |
| ID точки  перемещения  животного | INT | n N:(n > 0) | Первичный |
| ID локации | INT | n N:(n > 0) | Внешний |
| ID животного | INT | n N:(n > 0) | Внешний |
| Дата | DATE |  |  |

x

# Реализация веб-приложения

## Описание разделов веб-приложения

Интерфейс представляет собой формы стандартный интерфейс для программ. При проектировании главного меню необходимо соблюдать лаконичный дизайн, не перегруженное меню, а также чтобы пользоваться программой было интуитивно понятно.

Согласно общим требованиям, графический интерфейс разрабатываемого приложения должен:

* ориентироваться на пользователя, который общается с программой на внешнем уровне взаимодействия;
* сохранять стандартизированное назначение и местоположение на экране графических объектов, работающих в среде Windows.

Интерфейс программного приложения будет разрабатываться с учетом общих требований к пользовательскому интерфейсу.

Существуют общие требования, которые предъявляются к программным приложениям:

* соответствие стандартам организации интерфейса: использование многооконного подхода, реализация управления работой программного приложения с помощью элементов управления;
* требования к выбранной цветовой схеме рабочего экрана;
* выполнение одной функции с помощью разных элементов управления.

При запуске программы вы попадаете в главное меню, при помощи которого вы можете управлять приложением. В данном приложении главное окно должно состоять из следующих элементов меню:

* аутентификация;
* товары;
* корзина;
* панель администратора
* о программе;
* заказы.

Для удобства работы пользователя с программным средством необходимо при разработке форм придерживаться единого стиля оформления. Формы не должны быть перегружены излишней информацией или содержать информацию, не относящуюся к данной форме. Также необходимо предусмотреть защиту данных от удаления и изменения, а также от ввода некорректных данных. В случае ввода некорректных данных или попытке совершить запрещенные действия, пользователь должен быть проинформирован о своих действиях с помощью диалоговых окон.

## Разработка административной части приложения

x

## Разработка клиентской части приложения

x

## Описание используемых функций и процедур

x

## Функциональное тестирование

x

При разработке форма необходимо сохранять лаконичность и легкость интерфейса. Интерфейс должен выглядеть строго и интуитивно понятно. Интерфейс должен состоять из гармонирующих цветов, которые сочетаются друг с другом.

Концептуальный прототип состоит из описания внешнего пользовательского интерфейса, а именно, элементов управления.

При создании данного приложения важную роль играют формы, так как они являются основным средством работы с пользователем. Разрабатываемое приложение будет содержать несколько форм.

Работа приложения будет начинаться с главной формы, которая служит для основной работы с данным приложением. Главная форма будет содержать в себе все необходимые элементы управления.

Главная страница должна иметь в себе:

* аутентификация;
* товары;
* корзина;
* о программе;

Прототип главного окна изображен на рисунке 3.

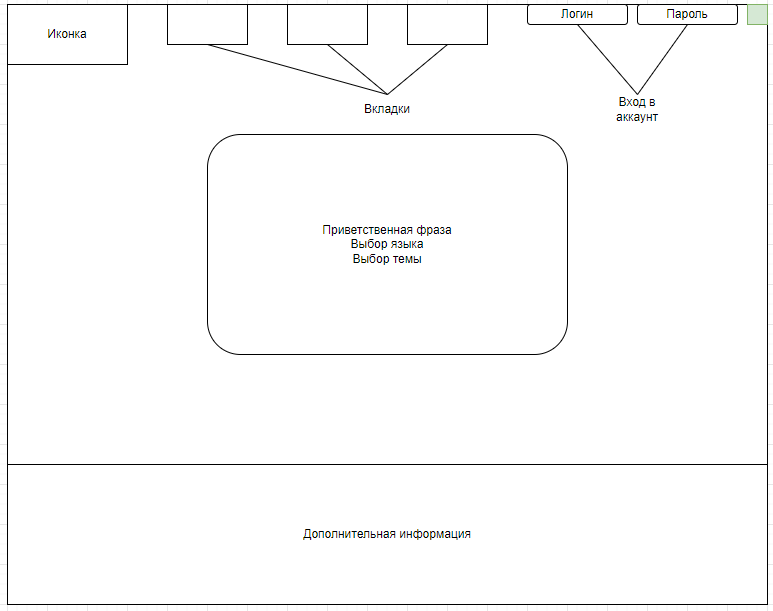


Рисунок 3 – Прототип главного окна

В таблице 1 представлена организация данных в базе данных «SQLite 3».

В таблице 2 представлены функции и закреплённые за ними элементы управления.

Таблица 2 – Функции и элементы управления

|  |  |  |  |
| --- | --- | --- | --- |
| Функция | Компонент меню | Название элемента  интерфейса | Реализация |
| Выход из  программы | btnClose\_Click | Щелчок мыши по кнопке «Профиль – Выход» | Средства С# |
| Авторизация | btnSignIN\_Click | Щелчок мыши по кнопке «Профиль – Войти» | Средства С# |
| Регистрация | btnSignUP\_Click | Щелчок мыши по кнопке «Профиль – Регистрация» | Средства С# |
| Панель администратора | btnAdmin\_Click | Щелчок мыши по кнопке «Панель администратора» | Средства С# |
| Заказы | btnOrder\_Click | Щелчок мыши по кнопке «Заказы» | Средства С# |
| Корзина | btnBasket\_Click | Щелчок мыши по кнопке «Корзина» | Средства С# |
| Профиль | UserPicture\_Click | Щелчок мыши по картинке профиля | Средства С# |
| О программе | gunaButton1\_Click | Щелчок мыши по картинке вопроса | Средства С# |
| Закрытие формы | btnCancel\_Click |  |  |
| Добавление  картинки | btnBrowse\_Click | Щелчок мыши по кнопке «Добавить картинку» | Средства С# |
| Добавление  товара | btnAddProduct\_Click | Щелчок мыши по кнопке «Добавить» | Средства С# |

В данном программной средстве справочная система, представленная в виде оконной формы с текстом информации о программе и ее пользовании, изображенной на рисунке x.

В данной справочной системе имеется шесть разделов, в которых описывается тема программы, её возможности, правила пользования с основным окном, мини-чаво, а также лицензия и авторы. Данная справочная системы вызывается путем нажатия на кнопку «О программе», которая расположенная в левом углу программы при нажатии на картинку вопроса.

Справочная система необходима для ознакомления с программным средством. В ней будет также изложена информация, которая может пригодиться пользователю. Содержание справочной системы должно быть просто и ясно изложено. Справочная система данного программного средства будет также содержать следующие разделы:

* «авторизация»;
* «главное меню»;
* «панель администратора».

На таблице 3 представлены требования к документированию.

Таблица 3 – Требования к документированию

|  |  |
| --- | --- |
| Проектирование. Разработка эскизного проекта. Разработка технического проекта. | Ведомость технического проекта |
| Разработка рабочей документации. Адаптация программ | Описание программ  Текст программ  Руководства пользователя |
| Ввод в действие | Акт завершения работ |

# Применение

## Назначение веб-приложения

Детальные сведения о программе и проблеме, которую она решает, приводятся в подразделе 1.1.

## Программно-аппаратное обеспечение сервера и клиента

x

# Методика испытаний

## Технические требования

Для полноценного функционирования разработанного программного приложения, качественного выполнения всех поставленных задач, требуется ПК следующей конфигурации:

* процессор с частотой не менее 1 ГГц;
* ОЗУ от 1 ГБ (для 32-разрядной ОС) или 2 ГБ (для 64-разрядной ОС);
* от 16 (для 32-разрядной ОС) или от 20 ГБ (для 64-разрядной ОС) свободного места на жестком диске;
* наличие DirectX 9 и более поздняя версия с драйвером WDDM 1.0;
* .NET Framework 4.0 или новее;
* операционная система: MS Windows 7 или новее.

Тестирование проводилось на аппаратных базах ноутбуков Asus N750JK и HP Pavilion 15, а также на стационарных ПК с различными техническими конфигурациями. Во время тестирования программы проблем, связанных с аппаратной совместимостью, выявлено не было.

## Функциональное тестирование

В этом подразделе приведены примеры реакций программы на ввод некорректных данных. При обнаружении ошибки отображается модальное окно с информацией о ней, а ввод некорректных данных отменяется. Это предохраняет пользователя от записи этих данных в программу и позволяет продолжить ее выполнение без прерываний и критических ошибок.

В таблицах 4-9 представлены методы тестирования «Use-case».

Таблица 4 – Use-case для авторизации

|  |  |
| --- | --- |
| ID | UC 4 |
| Цель | Выполнить авторизацию в приложении |
| Участники | Пользователь |
| Предусловие | Пользователь нажимает на кнопку «Аккаунт уже имеется?» |
| Триггер | Окно входа в приложение |
| Основной сценарий | 1. Пользователь вводит логин и пароль;  2. Пользователь запускает проверку;  3. Приложение проверяет логин;  4. Приложение проверяет пароль;  5. Приложение предоставляет пользователю доступ. |
| Альтернативный сценарий | 3.1 Не найдена учетная запись с таким логином;  3.1.1 Приложение уведомляет об ошибке;  3.2 Возврат сценария на пункт 1.  4.1 Пароль неверный;  4.1.1 Приложение уведомляет об ошибке;  4.2 Возврат сценария на пункт 1. |
| Исключения | Отсутствуют |

Таблица 5 – Use-case для регистрации

|  |  |  |
| --- | --- | --- |
| ID | UC 5 | |
| Цель | Выполнить регистрацию в приложении | |
| Участники | Пользователь | |
| Предусловие | Пользователь загружает приложение | |
| Продолжение таблицы 5 | | |  |
| Триггер | Окно входа в приложение | |
| Основной сценарий | 1. Пользователь вводит логин и пароль; | |
|  | 2. Пользователь запускает проверку;  3. Приложение проверяет логин;  4. Приложение проверяет пароль;  5. Приложение проверяет соответствие вводных данных в форме с уже зарегистрированными данными в БД;  6. Приложение подтверждает правильность заполнения формы;  7. Приложение предоставляет пользователю доступ. | |
| Альтернативный сценарий | 3.1 Логин длиннее 20 символов;  3.1.1 Приложение уведомляет об ошибке;  3.2 Возврат сценария на пункт 1.  3.3 Логин короче 5 символов;  3.3.1 Приложение уведомляет об ошибке;  3.4 Возврат сценария на пункт 1.  4.1 Пароль длиннее 40 символов;  4.1.1 Приложение уведомляет об ошибке;  4.2 Возврат сценария на пункт 1.  4.3 Пароль короче 10 знаков и не имеет специальных символов;  4.3.1 Приложение уведомляет об ошибке;  4.4 Возврат сценария на пункт 1.  5.1 Такой логин уже имеется  5.1.1 Приложение сообщает о том, что пользователь с таким логином уже зарегистрирован в системе;  5.2 Возврат сценария в пункт 1.  6.1 Форма заполнена неверно;  6.1.1 Приложение сообщает о том, что форма заполнена неверно;  6.2 Возврат сценария в пункт 1. | |
| Исключения | Отсутствуют | |

Таблица 6 – Use-case для выбора языка

|  |  |
| --- | --- |
| ID | UC 6 |
| Цель | Выполнить изменение языка окон |
| Участники | Пользователь |
| Предусловие | Пользователь нажимает на кнопку «Меню» |
| Триггер | Вкладка из меню |
| Основной сценарий | 1. Пользователь нажимает на кнопку «Меню»;  2. Пользователь нажимает на кнопку «Язык»;  3. Пользователь выбирает язык из 3 предложенных (Русский, Белорусский, Английский);  4. Приложение обрабатывает команду;  5. Приложение меняет язык. |
| Альтернативный сценарий | 4.1 Выбранный язык уже установлен  4.1.1 Язык остаётся прежним |
| Исключения | Отсутствуют |

Таблица 7 – Use-case для чтения сведений о программе

|  |  |
| --- | --- |
| ID | UC 7 |
| Цель | Выполнить чтение справки |
| Участники | Пользователь |
| Предусловие | Пользователь нажимает на кнопку «О программе» |
| Триггер | Вкладка из меню |
| Основной сценарий | 1. Пользователь нажимает на кнопку «Меню»;  2. Пользователь нажимает на кнопку «О программе»;  3. Приложение обрабатывает команду;  4. Приложение показывает справку. |
| Альтернативный сценарий | Отсутствует |
| Исключения | Отсутствуют |

Таблица 8 – Use-case для перекидывания товаров в корзину

|  |  |
| --- | --- |
| ID | UC 5 |
| Цель | Выполнить перекидывания товаров в корзину |
| Участники | Пользователь |
| Предусловие | Пользователь нажимает на кнопку «В корзину» |
| Триггер | Вкладка из меню |
| Основной сценарий | 1. Пользователь нажимает на кнопку «В корзину»;  2. Приложение обрабатывает команду;  3. Приложение перекидывает товар в корзину. |
| Альтернативный сценарий | 2.1. Товар уже есть в корзине  2.1.1. Кнопка «В корзину» заменяется на «Уже в корзине» и действие по кнопке заменяется на «только чтение». |
| Исключения | Отсутствуют |

Таблица 9 – Use-case для принятия заказа

|  |  |
| --- | --- |
| ID | UC 6 |
| Цель | Выполнить принятие заказа |
| Участники | Менеджер |
| Предусловие | Менеджер нажимает на кнопку «Заказы» |
| Триггер | Вкладка из меню |
| Основной сценарий | 1. Менеджер нажимает на кнопку «Заказы»;  2. Менеджер нажимает на кнопку «Активные»;  3. Менеджер нажимает на кнопку «Принять заказ»;  4. Приложение обрабатывает команду;  5. Приложение выводит информацию о заказе. |
| Альтернативный сценарий | 4.1. Заказ уже принял другой менеджер  4.1.1. Высвечивается сообщение о том, что заказ уже находится в обработке, а также просит обновить  Страницу. |
| Альтернативный сценарий | 4.2. Заказ был отменён пользователем  4.2.1. Заказ переходит во вкладку «Отменённые», а также высвечивается сообщение об отмене товара, а также просит обновить страницу. |
| Исключения | Отсутствуют |

В таблице 10 представлен метод тестирования «Bug-report».

Таблица 10 – Баг-репорт

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Краткое описание | Подробное описание | Шаги воспроизведения | | | | | Ожидаемый результат | | | Фактический результат | | | Воспроизводимость | Важность | | Срочность | Симптом |
| 1 | Выполняется авторизация при незаполненных полях "Логин" и "Пароль" в оконной форме авторизации. | В оконной форме авторизации при незаполненных полях "Логин" и "Пароль" выполняется авторизация и происходит переход на главную оконную форму. Требование: UC1 | 1.Окрыть программное приложение  2. Нажать на кнопку "Авторизоваться" 3. Не заполнить поле "Логин" 4. Не заполнить поле "Пароль" 5. Нажать кнопку "Авторизоваться" | | | | | Отображается сообщение "Обязательные поля не заполнены!" и не выполняется авторизация. | | | Не отображается сообщение "Обязательные поля не заполнены!" и выполняется авторизация. | | | Всегда | Средняя | | Средняя | Нереализованная функциональность |
| 2 | | Происходит ошибка в занесении данных о товаре в первую базу данных. | В оконной форме для занесения информации о товаре в базу данных при неверно введенных данных в поля "(ID) Идентификационный номер товара(обязательно)", "Наименование", "Цена(обязательно)", "Категория", "Производитель", "Количество" Требование: UC2 | 1. Открыть программное приложение  2. Нажать на кнопку "Авторизация" 3. Заполнить поле "Логин" корректными данными 4. Заполнить поле "Пароль" корректными данными 5. Нажать на кнопку "Товары" 6. Заполнить требуемые поля. 7. Нажать на кнопку "Записать в базу данных" | | | | | Отображается сообщение "Вы неверно заполнили форму!" и не выполняется занесение данных в первую базу данных. | | | Не отображается сообщение "Вы неверно заполнили форму!", и после этого выполняется занесение данных в первую базу данных. | | | Всегда | Высокая | | Высокая | Нереализованная функциональность |
| 3 | | Проис- | В оконной | 1. Открыть | | | | | Отобража- | | | Не отобра- | | | Всегда | Вы- | | Вы- | Неож |
| Продолжение таблицы 10 | | | | | |  |  |  | |  | | |  | | |  |  | |  |  |
|  | ходит ошибка в занесении данных о товаре во вторую базу данных. | форме для занесения информации о товаре в базу данных при неверно введенных данных в поля "(ID) Идентификационный номер товара(обязательно)", "Наименование", "Цена(обязательно)", "Категория", "Производитель", "Количество" Требование: UC3 | программное приложение  2. Нажать на кнопку "Авторизация" 3. Заполнить поле "Логин" корректными данными 4. Заполнить поле "Пароль" корректными данными 5. Нажать на кнопку "Товары" 6. Заполнить требуемые поля. 7. Нажать на кнопку "Записать в базу данных" | | | | | ется сообщение "Вы неверно заполнили форму!" и не выполняется занесение данных во вторую базу данных. | | | жается сообщение "Вы неверно заполнили форму!", и после этого выполняется занесение данных во вторую базу данных. | | |  | | сокая | сокая | иданное поведение |
| 4 | В главном окне программы в меню отсутствует пункт меню "Справка" Требование: UC4 | 1. Открыть программное приложение  2. Нажать на кнопку "Авторизация" 3. Заполнить поле "Логин" корректными данными 4. Заполнить поле "Пароль" | 1. Открыть программное приложение  2. Нажать на кнопку "Авторизация" 3. Заполнить поле "Логин" корректными данными 4. | | | | | Выводится оконная форма "Справка" | | | Оконная форма не выводится | | | Всегда | | Высокая | Высокая | Неожиданное поведение |
|  |  | корректными данными 5. Нажать на кнопку "Справка" Требование: UC4 | Заполнить поле "Пароль" корректными данными 5. Нажать на кнопку "Справка" | | | | |  | | |  | | |  | |  |  |  |
| 5 | После входа в аккаунт при открытии настрое | После входа в аккаунт при открытии настроек и нажатии на | 1. Открыть программное приложение  2. Нажать на кнопку | | | | | Выход с аккаунта | | | Производится переброс на главную страницу | | | Иногда | | Низкая | Низкая | Неожиданное поведение |
| Продолжение таблицы 10 | | | | | | | | |  | |  |  | |  |  |  | |  |  |  |
|  | к и нажатии на кнопку "Выйти" система должна осуществить выход | кнопку "Выйти" система должна осуществить выход Требование: UC5 | "Авторизация" 3. Заполнить поле "Логин" корректными данными 4. Заполнить поле "Пароль" корректными данными 5. Перейти в пункт настройки и нажать на кнопку "Выйти" | | | | |  | | |  | | |  | |  |  |  |
| 6 | При добавлении одного и того же товара в корзину система не позволяет добавить его в корзину. | 1. Открыть каталог  2. Добавить определенный товар в корзину  3. Проверить корзину  Требование: UC6 | Добавление товара в корзину. | | | | | Отображается сообщение "Товар уже имеется в корзине". | | | Отображается сообщение "Товар уже имеется в корзине". | | | Иногда | | Средняя | Низкая | Нереализованная функциональность |

# Применение

## Назначение программы

Назначение программы приведено в подразделе 4.2.

## Условия применения

Данное программное приложение не обладает средствами, ограничивающими доступ к нему, а также средствами, запрещающих распространять программу. Для работы с программным средством пользователю нужна операционная система MS Windows.

Для программы не установлены ограничения.

Требования к техническим и программным ресурсам программы приведены в подразделе 2.1.

## Справочная система

Справочная система приводится в подразделе 3.5.

# Заключение

В ходе выполнении проекта было задействовано множество форумов, сайтов, интернет-ресурсов, справочных материалов, методических указаний для разработки программного средства «E-Automize.exe», для продажи товаров в категории «Электроника».

Изначально были трудности с выбором базы данных, было протестировано три базы данных, такие как «OleDB», «SQLite 3», «MS SQL Server», выбор остановился на «SQLite 3».

Для достижения целей проектирования были решены следующие задачи:

* определена вычислительная система, необходимая для создания программного средства;
* разработана физическая и логическая модель данных;
* по модели выполнено проектирование задачи;
* разработано программное средство;
* описано созданное программное средство;
* выбрана методика испытаний;
* описан процесс тестирования;
* приведены примеры области применения.

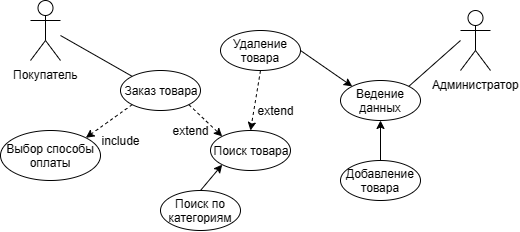
Разработка имеет интуитивно понятный графический интерфейс, позволяющий даже с минимальным знанием компьютера использовать данное программного средство.

Программа реализована в полном объеме и в соответствии с заданными требованиями, полностью отлажена и протестирована. Поставленные задачи выполнены.

К достоинствам данного программного средства можно отнести: интуитивно-понятный интерфейс, простоту в использовании.

В процессе разработки данного программного средства были применены и закреплены знания по уже изученному материалу, были отработаны навыки владения методами надежного программирования и эффективности разработки программного обеспечения в Microsoft Visual Studio 2022 с использованием языка программирования C#, разработана база данных средствами системы управления базами данных SQLite Studio.

В будущем будет реализована веб-версия, основной причиной является мобильность, не требуется установка «.exe» файла на компьютер, доступность из любого места, в котором имеется интернет-соединение, а также возможность интеграции со смартфоном.



*КБП*

*У*

*Т. Контр.*

*Разраб. Разраб.*

*Зеневич А.О.*

*Провер. Провер.*

*Купцова В.Ю.*

*Реценз.*

*Утверд.*

*Н. Контр.*

*Приложение по продаже товаров в категории «Электроника»*

*Диаграмма вариантов*

*использования*

*№ докум.*

*Подпись*

*Дата*

*Изм.*

*Лист*

*Масса*

*Лит.*

*Масштаб*

*Лист 1*

*Листов 5*

*Инв.№подл.*

*Подп. и дата*

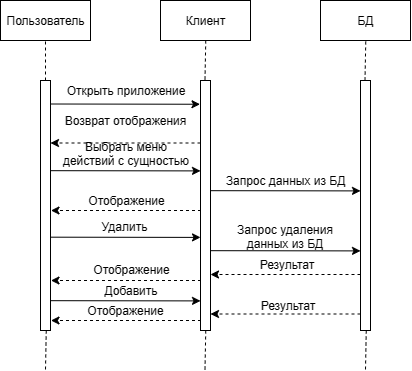
*Взам.инв.№*

*Инв.№дубл.*

*Подп. и дата*

*П Т.091005.401 ГЧ Т.ХХХХХХ.401 ГЧ*

*П Т.091005.401 ГЧ Т.ХХХХХХ.401 ГЧ*



*КБП*

*У*

*Т. Контр.*

*Разраб. Разраб.*

*Зеневич А.О.*

*Провер. Провер.*

*Купцова В.Ю.*

*Реценз.*

*Утверд.*

*Н. Контр.*

*Приложение по продаже товаров в категории «Электроника»*

*Диаграмма*

*последовательности*

*№ докум.*

*Подпись*

*Дата*

*Изм.*

*Лист*

*Масса*

*Лит.*

*Масштаб*

*Лист 2*

*Листов 5*

*Инв.№подл.*

*Подп. и дата*

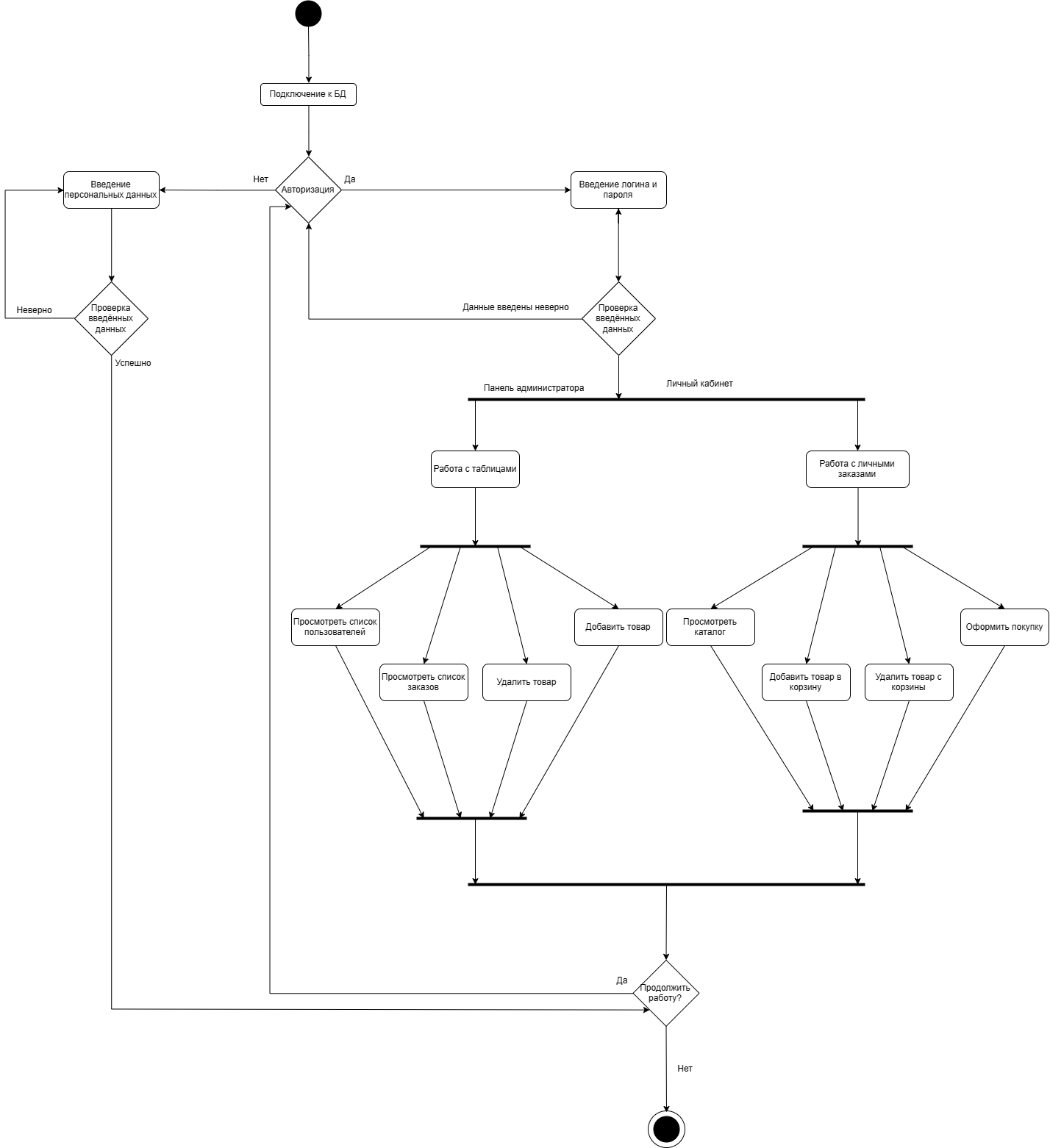
*Взам.инв.№*

*Инв.№дубл.*

*Подп. и дата*

*П Т.091005.401 ГЧ Т.ХХХХХХ.401 ГЧ*

*П Т.091005.401 ГЧ Т.ХХХХХХ.401 ГЧ*



*П Т.091005.401 ГЧ Т.ХХХХХХ.401 ГЧ*

*КБП*

*У*

*Т. Контр.*

*Разраб. Разраб.*

*Зеневич А.О.*

*Провер. Провер.*

*Купцова В.Ю.*

*Реценз.*

*Утверд.*

*Н. Контр.*

*Приложение по продаже товаров в категории «Электроника»*

*Диаграмма деятельности*

*№ докум.*

*Подпись*

*Дата*

*Изм.*

*Лист*

*Масса*

*Лит.*

*Масштаб*

*Лист 3*

*Листов 5*

*Инв.№подл.*

*Подп. и дата*

*Взам.инв.№*

*Инв.№дубл.*

*Подп. и дата*

*П Т.091005.401 ГЧ Т.ХХХХХХ.401 ГЧ*

*П Т.091005.401 ГЧ Т.ХХХХХХ.401 ГЧ*

*КБП*

*У*

*Т. Контр.*

*Разраб. Разраб.*

*Зеневич А.О.*

*Провер. Провер.*

*Купцова В.Ю.*

*Реценз.*

*Утверд.*

*Н. Контр.*

*Приложение по продаже товаров в категории «Электроника»*

*Диаграмма классов*

*№ докум.*

*Подпись*

*Дата*

*Изм.*

*Лист*

*Масса*

*Лит.*

*Масштаб*

*Лист 4*

*Листов 5*

*Инв.№подл.*

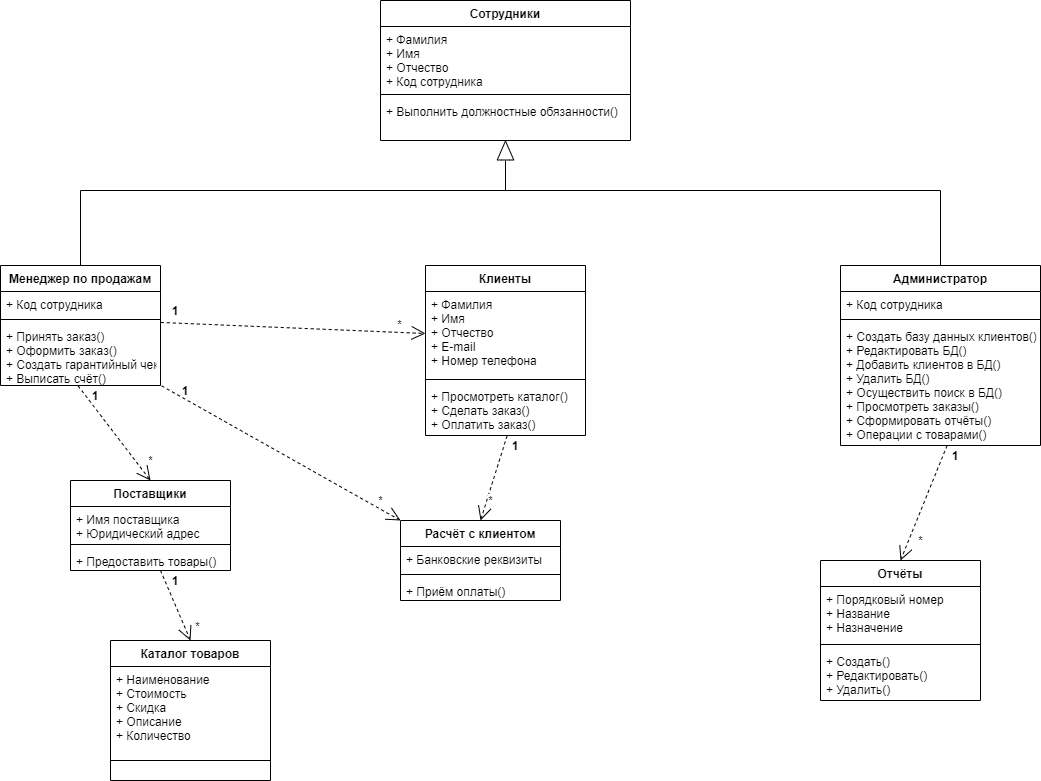
*Подп. и дата*

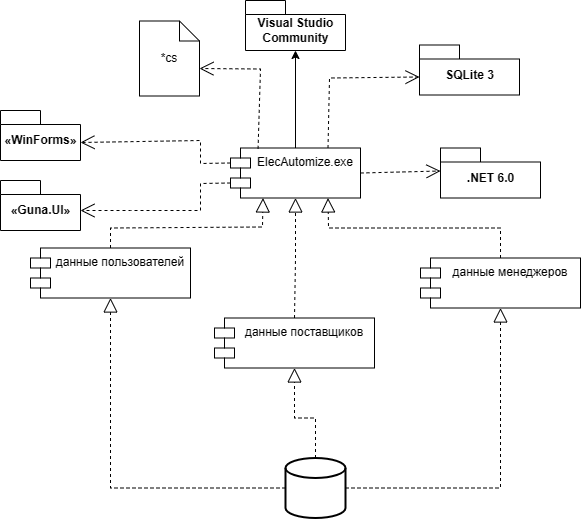
*Взам.инв.№*

*Инв.№дубл.*

*Подп. и дата*

*П Т.091005.401 ГЧ Т.ХХХХХХ.401 ГЧ*





users.db

*КБП*

*У*

*Т. Контр.*

*Разраб. Разраб.*

*Зеневич А.О.*

*Провер. Провер.*

*Купцова В.Ю.*

*Реценз.*

*Утверд.*

*Н. Контр.*

*Приложение по продаже товаров в категории «Электроника»*

*Диаграмма компонентов*

*№ докум.*

*Подпись*

*Дата*

*Изм.*

*Лист*

*Масса*

*Лит.*

*Масштаб*

*Лист 5*

*Листов 5*

*Инв.№подл.*

*Подп. и дата*

*Взам.инв.№*

*Инв.№дубл.*

*Подп. и дата*

*П Т.091005.401 ГЧ Т.ХХХХХХ.401 ГЧ*

*П Т.091005.401 ГЧ Т.ХХХХХХ.401 ГЧ*