**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 8](#_Toc195467229)

[1 Постановка задачи 9](#_Toc195467230)

[1.1 Анализ предметной области 9](#_Toc195467231)

[1.2 Техническое задание на разработку ГОСТ 19.201-78. 9](#_Toc195467232)

[1.2.1 Введение 9](#_Toc195467233)

[1.2.1.1 Наименование программы 9](#_Toc195467234)

[1.2.1.2 Краткая характеристика области применения 9](#_Toc195467235)

[1.2.2 Основания для разработки 9](#_Toc195467236)

[1.2.2.1 Основание для проведения разработки 9](#_Toc195467237)

[1.2.2.2 Наименование темы разработки 9](#_Toc195467238)

[1.2.3 Назначение разработки 9](#_Toc195467239)

[1.2.3.1 Функциональное назначение 9](#_Toc195467240)

[1.2.3.2 Эксплуатационное назначение 10](#_Toc195467241)

[1.2.4 Требования к программе или программному изделию 10](#_Toc195467242)

[1.2.4.1 Требования к функциональным характеристикам 10](#_Toc195467243)

[1.2.4.2 Требования к надежности 10](#_Toc195467244)

[1.2.4.3 Время восстановления после отказа 10](#_Toc195467245)

[1.2.4.4 Отказы из-за некорректных действий оператора 10](#_Toc195467246)

[1.2.4.5 Условия эксплуатации 11](#_Toc195467247)

[1.2.4.6 Требования к видам обслуживания 11](#_Toc195467248)

[1.2.4.7 Требования к информационной и программной совместимости 11](#_Toc195467249)

[1.2.4.8 Требования к программной документации 11](#_Toc195467250)

[1.2.5 Стадии и этапы разработки 12](#_Toc195467251)

[1.2.5.1 Стадии разработки 12](#_Toc195467252)

[1.2.5.2 Этапы разработки 12](#_Toc195467253)

[1.2.5.3 Содержание работ по этапам 12](#_Toc195467254)

[1.2.6 Порядок контроля и приёмки 13](#_Toc195467255)

[1.2.6.1 Виды испытаний 13](#_Toc195467256)

[2 Проектирование 14](#_Toc195467257)

[2.1 Функциональная структура программы 14](#_Toc195467258)

[2.2 Схема модулей программы 14](#_Toc195467259)

[2.3 Паспорта основных модулей 15](#_Toc195467260)

[2.3.1 Модуль btnAddProduct 15](#_Toc195467261)

[2.3.2 Модуль btnEditProduct 16](#_Toc195467262)

[2.3.3 Модуль btnDeleteProduct 17](#_Toc195467263)

[2.3.4 Модуль OnSearchProduct 18](#_Toc195467264)

[3 Тестирование программы 19](#_Toc195467265)

[3.1 Программа, методика и результаты испытаний ГОСТ 19.301-79 19](#_Toc195467266)

[3.1.1 Объект испытаний 19](#_Toc195467267)

[3.1.1.1 Наименование программы 19](#_Toc195467268)

[3.1.1.2 Назначение разработки 19](#_Toc195467269)

[3.1.2 Цель испытаний 19](#_Toc195467270)

[3.1.2.1 Основания для проведения испытаний 19](#_Toc195467271)

[3.1.2.2 Место и продолжительность испытаний 19](#_Toc195467272)

[3.1.2.3 Перечень документов, предъявляемых на испытании 19](#_Toc195467273)

[3.1.2.4 Объём испытаний 19](#_Toc195467274)

[3.1.2.4.1 Перечень проверок, проводимых на 1 этапе испытаний 19](#_Toc195467275)

[3.1.2.4.2 Перечень проверок, проводимых на 2 этапе испытаний 20](#_Toc195467276)

[3.1.2.4.3 Количественные и качественные характеристики, подлежащие оценке 20](#_Toc195467277)

[3.1.3 Требования к программе 20](#_Toc195467278)

[3.1.4 Требования к программной документации 20](#_Toc195467279)

[3.1.5 Средства и порядок испытаний 21](#_Toc195467280)

[3.1.5.1 Технические средства, используемые во время испытаний 21](#_Toc195467281)

[3.1.5.1.1 Порядок проведения испытаний 21](#_Toc195467282)

[3.1.5.2 Условия и порядок проведения испытаний 21](#_Toc195467283)

[3.1.5.2.1 Условия начала и завершения отдельных этапов испытаний 21](#_Toc195467284)

[3.1.5.2.2 Требования к техническому обслуживанию 21](#_Toc195467285)

[3.1.6 Методика испытаний 21](#_Toc195467286)

[3.2 Результаты предварительных испытаний 22](#_Toc195467287)

[4 Мобильное приложение с отображением главной таблицы 24](#_Toc195467288)

[4.1 Скриншот вывода информации 24](#_Toc195467289)

[4.2 Листинг 24](#_Toc195467290)

[4.2.1 Листинг Json 24](#_Toc195467291)

[4.2.2 Листинг интерфейса 25](#_Toc195467292)

[5 Рефакторинг кода 26](#_Toc195467293)

[ЗАКЛЮЧЕНИЕ 27](#_Toc195467294)

# **ВВЕДЕНИЕ**

# В современном мире цифровых технологий эффективная командная работа играет ключевую роль в успешном выполнении проектов, особенно в условиях удалённого взаимодействия и распределённых коллективов. Традиционные инструменты для совместной работы, такие как мессенджеры и таск-менеджеры, обеспечивают коммуникацию и управление задачами, но зачастую не предлагают интерактивных возможностей для творческого взаимодействия. В связи с этим актуальной становится разработка специализированных приложений, сочетающих в себе элементы визуального collaboration-инструментария и функционал для обмена файлами.

# Данная работа посвящена созданию клиентской части приложения для командной работы, которое объединяет в себе функционал графического редактора (аналогичного работе с маркерной доской) с возможностью многопользовательского взаимодействия в реальном времени, а также инструменты для размещения и совместного использования файлов. Подобное решение позволит командам не только обсуждать задачи, но и визуализировать идеи, вносить правки в графические материалы и оперативно обмениваться документами в едином пространстве.

# В процессе создания приложения будут использованы следующие инструменты и технологии:

1. Язык программирования: JavaScript
2. Библиотеки пользовательского интерфейса: React, Konva

# **1 Постановка задачи**

# **1.1 Анализ предметной области**

Цель разработки информационной системы (ИС) «Приложение для командной работы» заключается в создании универсального инструмента, который позволит эффективно управлять процессами разработки, внедрения и поддержки программного обеспечения. Это будет способствовать улучшению качества программных решений, оптимизации рабочего процесса и максимизации удовлетворенности пользователей.

# **1.2 Техническое задание на разработку ГОСТ 19.201-78.**

# **1.2.1 Введение**

# **1.2.1.1 Наименование программы**

Наименование программы «BrainStorm».

# **1.2.1.2 Краткая характеристика области применения**

Программа предназначена для ускорения процессов кооперативного принятия решений и планирования в бизнесе.

# **1.2.2 Основания для разработки**

# **1.2.2.1 Основание для проведения разработки**

Основанием для проведения разработки является задание на производственную преддипломную практику по модулю ПМ.01. Разработка модулей программного обеспечения для компьютерных систем.

# **1.2.2.2 Наименование темы разработки**

Наименование темы разработки: «Проектирование информационной системы «Приложение для командной работы».

# **1.2.3 Назначение разработки**

# **1.2.3.1 Функциональное назначение**

Функциональное назначение информационной системы (ИС) "Приложение для командной работы" представляет собой интерактивную платформу для совместной работы, сочетающую функции графического редактора с многопользовательским режимом и возможностью обмена файлами. Оно предназначено для команд, которым необходимо не только обсуждать задачи, но и визуализировать идеи, оперативно вносить правки в графические материалы и обмениваться документами в режиме реального времени.

# **1.2.3.2 Эксплуатационное назначение**

Требования эксплуатационного назначения не предъявляются.

# **1.2.4 Требования к программе или программному изделию**

# **1.2.4.1 Требования к функциональным характеристикам**

Программа должна обладать следующими функциональными характеристиками:

* функция входа в аккаунт,
* функция выхода из аккаунта,
* функция создания комнаты,
* функция выхода комнаты,
* функция рисования на холсте,
* функция изменения параметров компонента на холсте,
* функция удаления компонента на холсте,
* функция синхронизации компонентов,
* функция перемещения по холсту,
* функция загрузки изображения на холст.

# **1.2.4.2 Требования к надежности**

Требования к обеспечению надежного функционирования программы не предъявляются.

# **1.2.4.3 Время восстановления после отказа**

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 1 часа для пользовательского интерфейса и 24 часов для бизнес-логики при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

# **1.2.4.4 Отказы из-за некорректных действий оператора**

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу пользователя без предоставления ему административных привилегий.

# **1.2.4.5 Условия эксплуатации**

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

# **1.2.4.6 Требования к видам обслуживания**

Программа не требует проведения каких-либо видов обслуживания.

# **1.2.4.7 Требования к информационной и программной совместимости**

**Требования к информационным структурам и методам решения**

Информационная структура приложения должна обеспечивать:

1. Отображение графического холста с инструментами рисования в режиме реального времени.
2. Поддержку загружаемых файлов (изображения, документы) с возможностью последующей выгрузкой файла на устройство пользователя.
3. Синхронизацию действий пользователей при совместном редактировании и исключения коллизий данных.

**Требования к исходным кодам и языкам программирования**

Исходные коды программы должны быть реализованы на языках JavaScript. В качестве интегрированной среды разработки программы должна быть использована среда Visual Studio Code.

**Требования к программным средствам, используемым программой**

Системные программные средства, используемые конечным пользователям, должны быть представлены лицензионной локализованной версией любой операционной системы с браузером на базе Chrome версии 100 и новее.

Системные программные средства, используемые для обработки данных и размещения веб-сервера, должны быть представлены лицензионной локализованной версией Debian 9 и новее, веб-сервером nginx версии 1.28 и новее, брокером сообщений RabbitMQ версии 4.0.0 и новее.

**Требования к защите информации и программ**

Информация, передаваемая между клиентом и сервером должна быть передана с использованием протокола SSL и шифрования.

**Специальные требования**

Программа должна обеспечивать взаимодействие с пользователем посредством графического пользовательского интерфейса, разработанного согласно рекомендациям и современным тенденциям к разработке веб-интерфейсов.

# **1.2.4.8 Требования к программной документации**

Предварительный состав программной документации

Состав программной документации должен включать в себя:

* техническое задание на разработку ГОСТ 19.2010-78,
* паспорта модулей.

**1.2.5 Стадии и этапы разработки**

# **1.2.5.1 Стадии разработки**

Разработка должна быть проведена в две стадии:

* техническое задание,
* технический и рабочий проекты.

# **1.2.5.2 Этапы разработки**

На стадии «Техническое задание» должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии «Технический и рабочий проекты» должны быть выполнены перечисленные ниже этапы работ:

* разработка программы,
* разработка программной документации,
* испытания программы.

# **1.2.5.3 Содержание работ по этапам**

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

* постановка задачи,
* определение и уточнение требований к техническим средствам,
* определение требований к программе,
* определение стадий, этапов и сроков разработки программы и документации на неё,
* выбор языка программирования,
* согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями ГОСТ.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

* разработка, согласование и утверждение программы и методики испытаний,
* проведение приёмосдаточных испытаний,
* корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по сдаче программы и программной документации преподавателю.

# **1.2.6 Порядок контроля и приёмки**

# **1.2.6.1 Виды испытаний**

Приёмосдаточные испытания должны проводиться на объекте заказчика в сроки выполнения для курсового проекта.

Приёмосдаточные испытания программы должны проводиться согласно разработанной исполнителем и согласованно заказчиком «Программы и методики испытаний».

# **2 Проектирование**

# **2.1 Функциональная структура программы**

Таблица 1 Спецификация функций программы

|  |  |  |  |
| --- | --- | --- | --- |
| **Название** | **Действие** | **Объект** | **Вид** |
| **Авторизация** | | | | |
| Вход в аккаунт | Вход в существующий аккаунт пользователя | Пользователь | Основная |
| Выход из аккаунта | Выход из текущего аккаунта пользователя | Пользователь | Основная |
| **Аккаунт** | | | |
| Создание комнаты | Создание новой комнаты для коллабораций | Название комнаты | Основная |
| Удаление комнаты | Удаление существующей комнаты | Комната | Основная |
| **Холст** | | | |
| Перо | Рисование на холсте | Холст | Основная |
| Рука | Выбор элемента на холсте, трансформация, удаление | Объект | Основная |
| Табличка | Добавление текста на холст | Холст | Основная |
| Картинка | Добавление картинки на холст | Холст | Основная |
| Панель свойств | Редактирование свойств компонентов | Холст | Вспомогательная |
| Панель инструментов | Отображение доступных инструментов | Холст | Вспомогательная |
| **Глобальные** | | | |
| «Тостер» | Показ системных уведомлений | «Тост» | Вспомогательная |
| Сервис авторизации | Взаимодействие с API авторизации | Данные пользователя | Вспомогательная |
| Сервис синхронизации | Взаимодействие с WebSocket для синхронизации действий | Данные сессии | Вспомогательная |

# **2.2 Схема модулей программы**

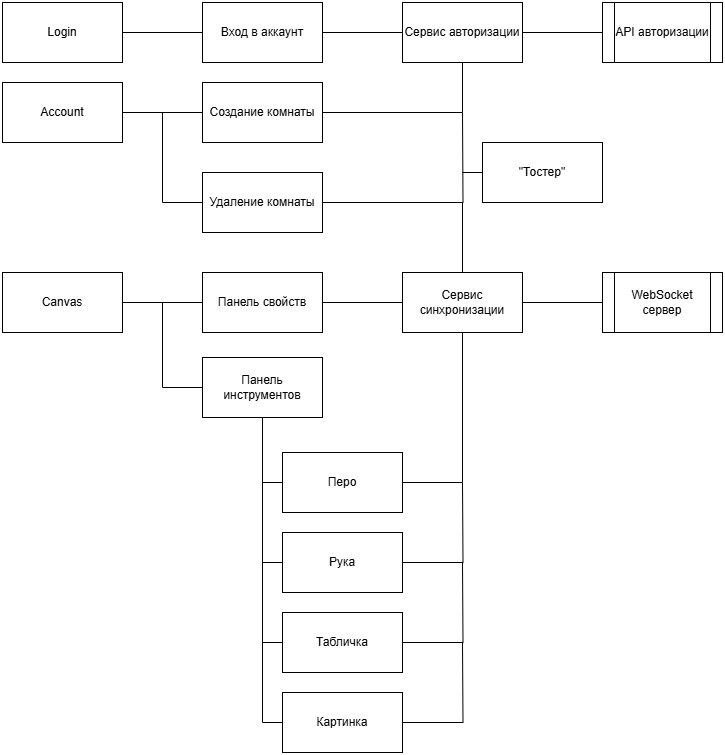


Рисунок 1 – Модульная структура программы

# **2.3 Паспорта основных модулей**

Примечание к паспортам модулей: в связи с особенностями разработки с использованием фреймворка React, блок схемы будут отражать общую суть логики действий программы.

# **2.3.1 Модуль btnAddProduct**

Таблица 2 Спецификация модуля btnAddProduct

|  |  |  |
| --- | --- | --- |
| **Имя** | **Реализуемая функция** | **Параметры** |
| void btnAddProduct (string errors, ListView  lv\_ Product) | Вывод страницы авторизации | Входные данные: errors – переменная для передачи ошибок при редактировании списка  Выходные данные: lv\_ Product - добавление записи в список |

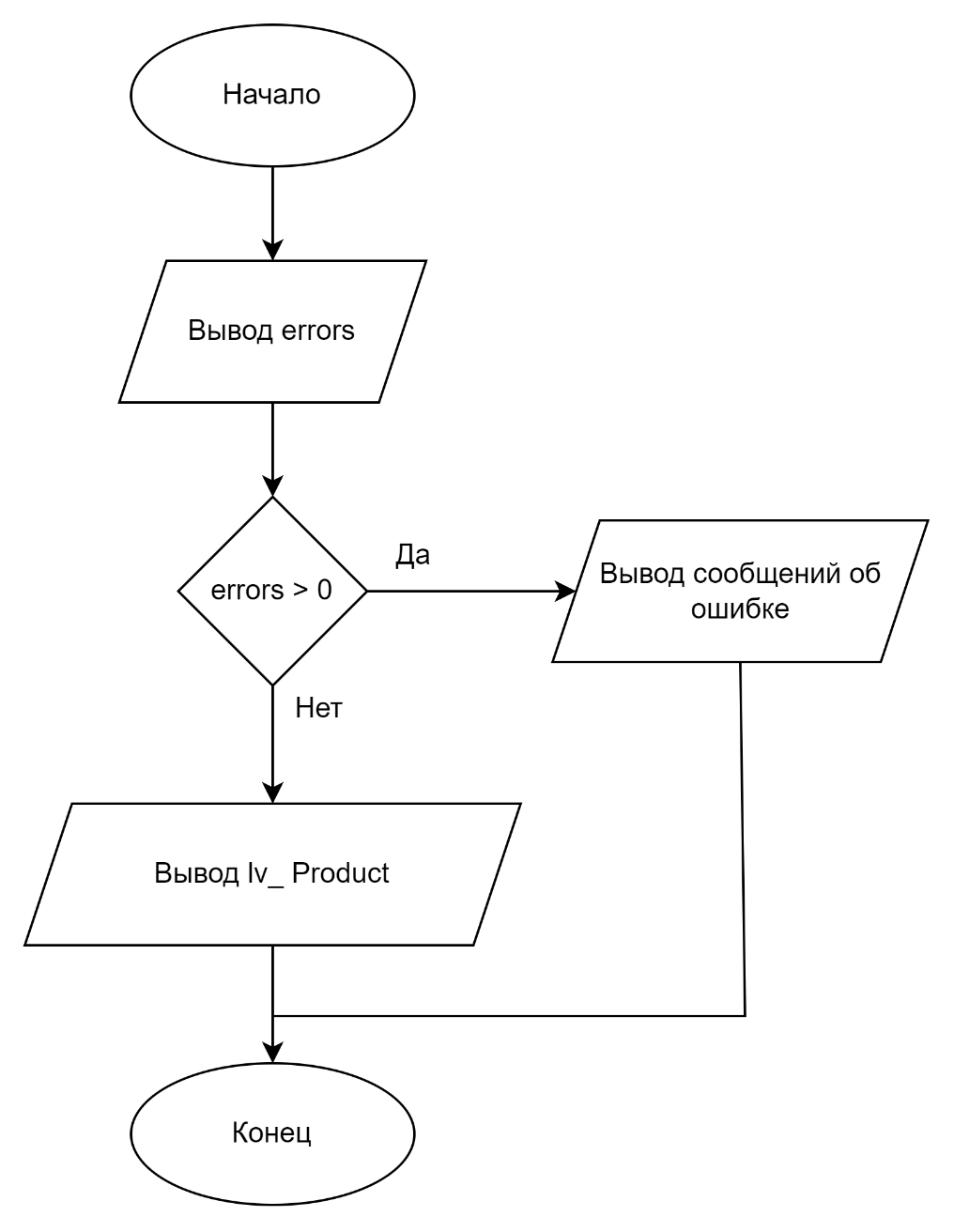


Рисунок 2 – Алгоритм добавления записи

# **2.3.2 Модуль btnEditProduct**

Таблица 3 Спецификация модуля btnEditProduct

|  |  |  |
| --- | --- | --- |
| **Имя** | **Реализуемая функция** | **Параметры** |
| void btnEditProduct (string errors, ListView  lv\_ Product) | Редактирование записи в список | Входные данные: errors – переменная для передачи ошибок при редактировании списка  Выходные данные: lv\_ Product – добавление измененной записи в список |

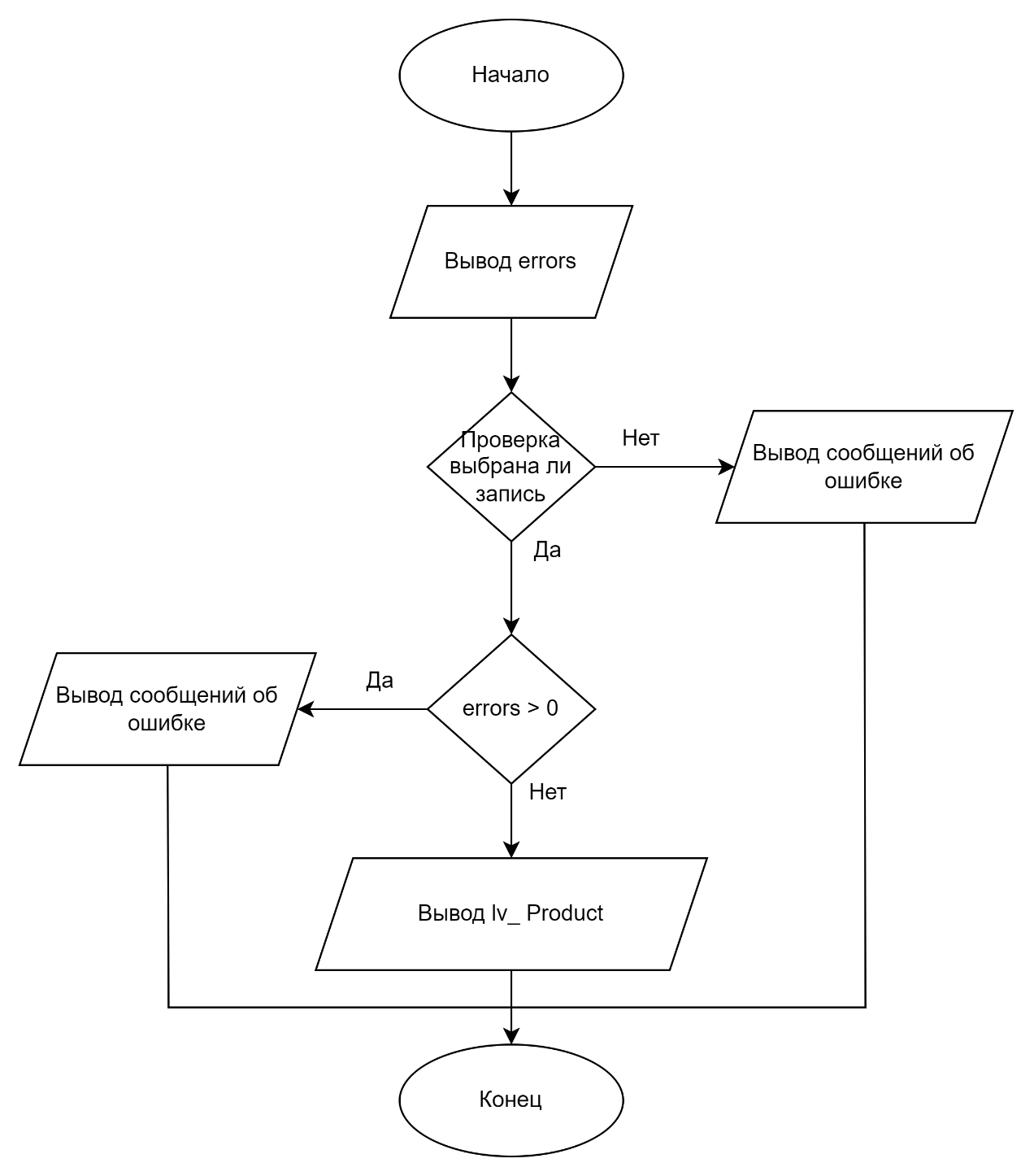


Рисунок 3 – Алгоритм редактирования записи

# **2.3.3 Модуль btnDeleteProduct**

Таблица 4 Спецификация модуля btnDeleteProduct

|  |  |  |
| --- | --- | --- |
| **Имя** | **Реализуемая функция** | **Параметры** |
| void btnDeleteProduct (bool result, productToDelete) | Удаление записи из списка | Входные данные: result – переменная для передачи ошибок при удалении элемента  Выходные данные: productToDelete – параметр, содержащий результат операции удаления |

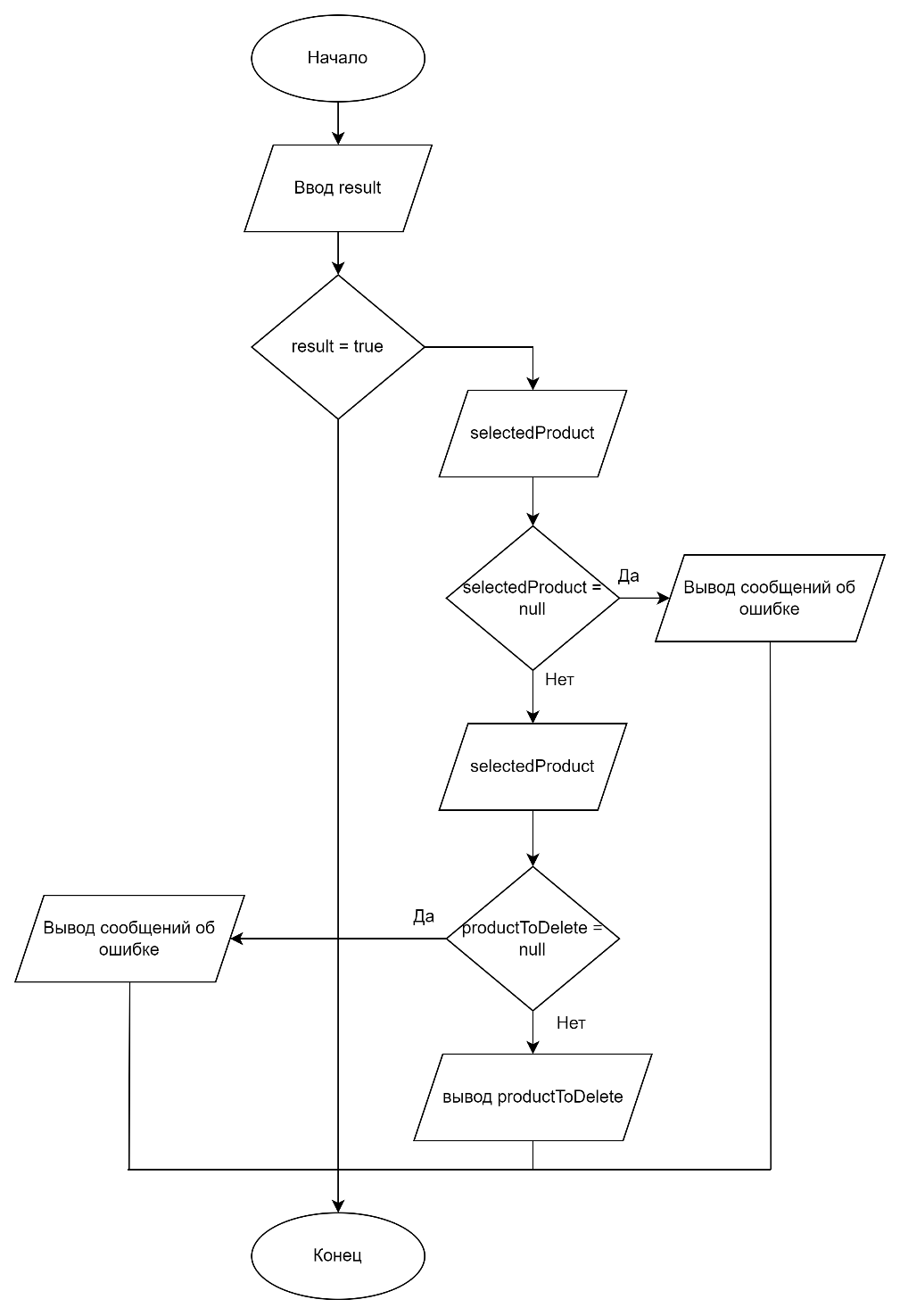


Рисунок 4 – Алгоритм удаления записи

# **2.3.4 Модуль OnSearchProduct**

Таблица 5 Спецификация модуля OnSearchProduct

|  |  |  |
| --- | --- | --- |
| **Имя** | **Реализуемая функция** | **Параметры** |
| void OnSearchProduct (object sender, TextChangedEventArgs e) | Поиск в списке. | sender – объект – инициатор события; e – аргумент события типа TextChangedEventArgs |

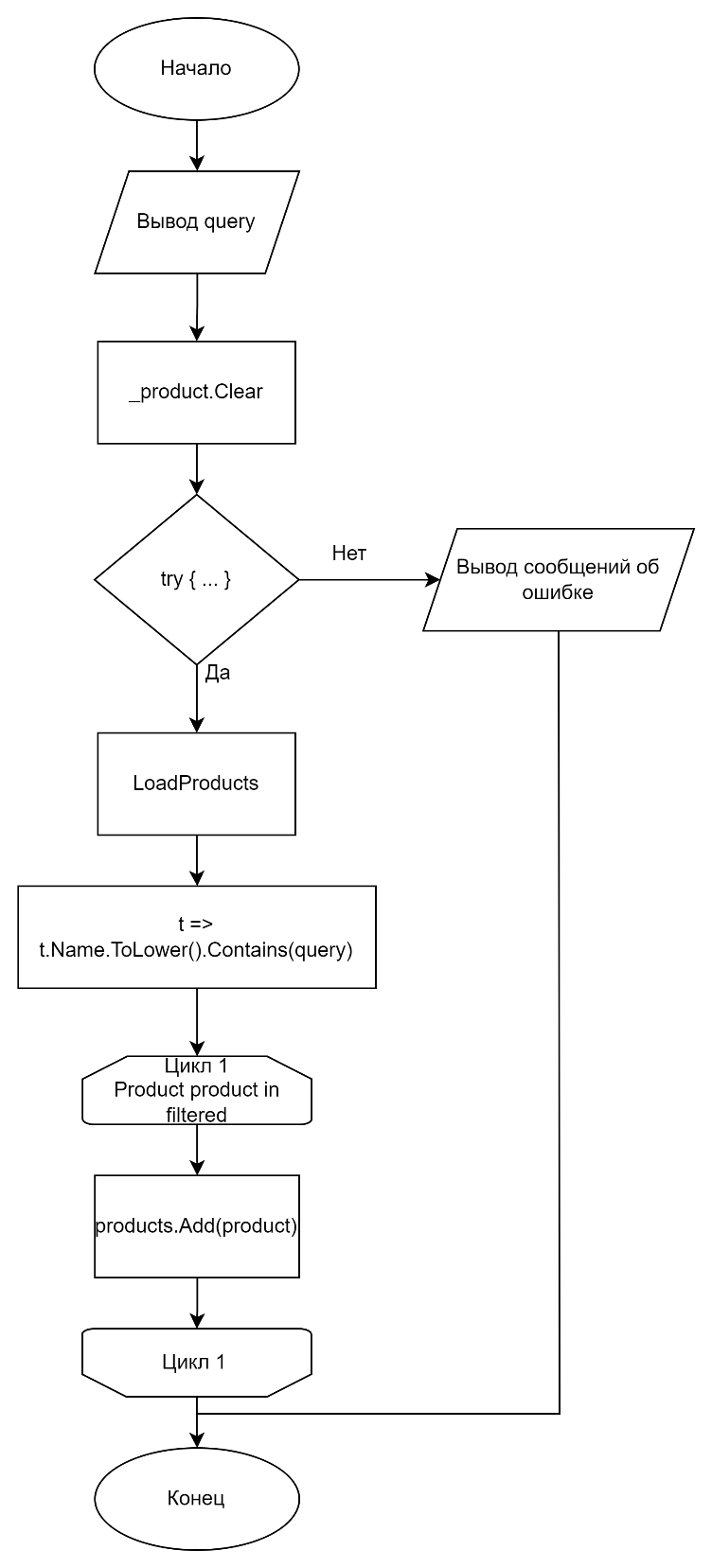


Рисунок 5 – Алгоритм поиска

# **3 Тестирование программы**

# **3.1 Программа, методика и результаты испытаний ГОСТ 19.301-79**

# **3.1.1 Объект испытаний**

# **3.1.1.1 Наименование программы**

Наименование программы «SoftwareProducts».

# **3.1.1.2 Назначение разработки**

Программа предназначена для автоматизации управления программными продуктами.

# **3.1.2 Цель испытаний**

Цель проведения испытаний – проверка соответствия характеристик разработанной программы (программного изделия) функциональным и отдельным иным видам требований, изложенным в документе «Техническое задание».

# **3.1.2.1 Основания для проведения испытаний**

Основанием является полное выполнение задания на производственную практику.

# **3.1.2.2 Место и продолжительность испытаний**

Местами проведения испытаний является дом и место прохождения практики. Продолжительность самих испытаний 1 неделя.

# **3.1.2.3 Перечень документов, предъявляемых на испытании**

Состав программной документации должен включать в себя:

* техническое задание,
* программа, методика и результаты испытаний.

# **3.1.2.4 Объём испытаний**

Испытаний проводятся в два этапа, а именно ознакомительный и испытания.

# **3.1.2.4.1 Перечень проверок, проводимых на 1 этапе испытаний**

Перечь проверок, проводимых на 1-м этапе испытаний, должен включать в себя:

* проверку комплектности программной документации,
* проверку комплектности состава технических и программных средств.

# **3.1.2.4.2 Перечень проверок, проводимых на 2 этапе испытаний**

Перечь проверок, проводимых на 2-м этапе испытаний, должен включать в себя:

* проверку соответствия технических характеристик программы,
* проверку степени выполнения требований функционального назначения программы.

# **3.1.2.4.3 Количественные и качественные характеристики, подлежащие оценке**

В ходе проведения приемосдаточных испытаний оценке подлежат качественные (функциональные) характеристики программы. Проверке подлежит возможность выполнения программой перечисленных ниже функций:

* функция добавления,
* функция редактирования,
* функция удаления,
* функция поиска.

# **3.1.3 Требования к программе**

При проведении испытаний функциональные характеристики (возможности) программы подлежат проверке на соответствие требованиям, изложенным в п. «Требования к функциональным выполняемых функций» технического задания.

# **3.1.4 Требования к программной документации**

Состав программной документации должен включать в себя:

* техническое задание,
* программа, методика и результаты.

# **3.1.5 Средства и порядок испытаний**

# **3.1.5.1 Технические средства, используемые во время испытаний**

В состав технических средств должен входить IBM-совместимый персональный компьютер (ПЭВМ), включающая в себя:

* процессор: AMD Ryzen 3,
* ОЗУ: 4Гб и более,
* необходимое место на жёстком диске: 1Гб,
* видеоадаптер: AMD Radeon Vega,
* экран: 1280 х 960.

# **3.1.5.1.1 Порядок проведения испытаний**

Испытания должны проводиться поэтапно согласно п. «Перечень этапов испытаний» настоящего документа.

# **3.1.5.2 Условия и порядок проведения испытаний**

# **3.1.5.2.1 Условия начала и завершения отдельных этапов испытаний**

Необходимым и достаточным условием завершения 1 этапа испытаний и начала 2 этапа испытаний является успешное завершение проверок, проводимых на 1 этапе (см. п. «Перечень проверок, проводимых на 1 этапе испытаний»).

Условием завершения 2 этапа испытаний является успешное завершение проверок, проводимых на 2 этапе испытаний.

# **3.1.5.2.2 Требования к техническому обслуживанию**

Требования к техническому обслуживанию не предъявляются.

# **3.1.6 Методика испытаний**

**Методы проведения проверки комплектности программной документации**

Проверка комплектности программной документации на программное изделие производится визуально преподавателем. В ходе проверки сопоставляется состав и комплектность программной документации, представленной исполнителем, с перечнем программной документации.

Проверка считается завершенной в случае соответствия состава и комплектности программной документации, представленной исполнителем, перечню программной документации, приведенному в указанном выше пункте.

**Методы проведения проверки степени выполнения требований функционального назначения программы**

Тестирование черного ящика, также известное как поведенческое тестирование, представляет собой методологию, направленную на оценку функциональности программного обеспечения, сосредоточив внимание на его внешних характеристиках и поведении. В этом подходе тестировщик не учитывает внутреннюю архитектуру или код приложения, а вместо этого ориентируется на спецификации и требования, описанные для системы.

# **3.2 Результаты предварительных испытаний**

Результаты предварительных испытаний черным ящиком будут отображены ниже в таблице 6.

Таблица 6 Функциональное тестирование «черного ящика»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Тестовая спецификация** | | | | | | |
| **Разработано:** Юсубов Рустам | | | **Начальные установки:** отсутствуют | | | |
| **Цель:** проверки степени выполнения требований функционального назначения программы. | | | | | | |
| **Test case**  **#** | **Описание** | **Шаги теста** | **Ожидаемый результат** | **Фактический результат** | **Пройден/не пройден** | **Тестер**  **/Дата** |
| **Тестирование основных функций** | | | | | | |
| 1 | Тестирование функциональности добавления записи в список | Нажать на кнопку «Добавить» | Добавление новой записи в список | Добавление новой записи в список | Пройден | Сафронова Юлия 19.03.2025 |

Продолжение таблицы 6

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | Тестирование функциональности добавления пустой записи в список | | Нажать на кнопку «Добавить», оставить поля пустыми, нажать на кнопку «Добавить» | | Ошибка добавления записи | | Ошибка добавления записи | | Пройден | Сафронова Юлия 19.03.2025 |
| 3 | Тестирование функциональности редактирования записи в списке | Выбрать запись и нажать на кнопку «Изменить» | | Измененная информация сохраняется в списке | | Измененная информация сохраняется в списке | | Пройден | | Сафронова Юлия 19.03.2025 |
| 4 | Тестирование функциональности удаления записи в списке | Выбрать запись и нажать на кнопку «Удалить» | | Выбранная запись удаляется из списка | | Выбранная запись удаляется из списка | | Пройден | | Сафронова Юлия 19.03.2025 |
| 5 | Тестирование функциональности поиска | Пишем в SearchBar ключевое слово, которое не содержится в таблице | | Совпадений не найдено | | Совпадений не найдено | | Пройден | | Сафронова Юлия 19.03.2025 |
| 6 | Тестирование функциональности поиска | Пишем в SearchBar ключевое слово, которое содержится в таблице | | В таблице должны отобразиться все записи, которые совпадают с введённым в поиск, словом, | | В таблице должны отобразиться все записи, которые совпадают с введённым в поиск, словом, | | Пройден | | Сафронова Юлия 19.03.2025 |

# **4 Мобильное приложение с отображением главной таблицы**

# **4.1 Скриншот вывода информации**

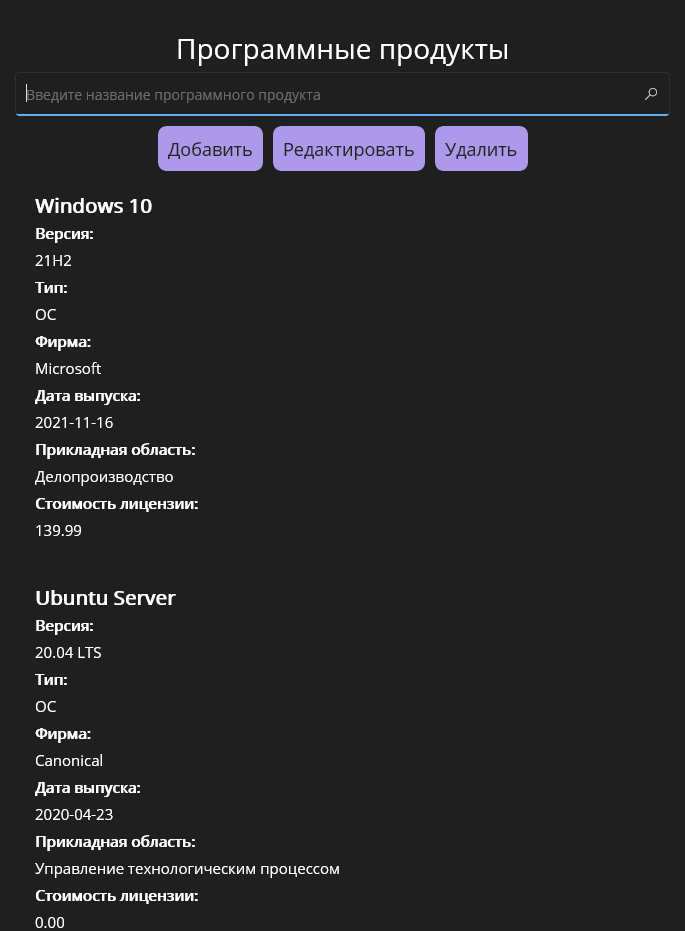


Рисунок 6 – Вывод списка

# **4.2 Листинг**

# **4.2.1 Листинг Json**

class APIMetodsl

{

const string Url = "http://localhost:500/api/ Product";

public static T Get<T>()

{

WebClient client = new WebClient();

client.Encoding = Encoding.UTF8;

var response = client.DownloadString(Url);

return JsonConvert.DeserializeObject<T>(response);

}

public static string Post<T>(T body)

{

WebClient client = new WebClient();

client.Encoding = Encoding.UTF8;

client.Headers.Add(HttpRequestHeader.ContentType, "application/json");

string data = JsonConvert.SerializeObject(body);

return client.UploadString(Url, "POST", data);

}

public static string Put<T>(T body, int id)

{

WebClient client = new WebClient();

client.Encoding = Encoding.UTF8;

client.Headers.Add(HttpRequestHeader.ContentType, "application/json");

string data = JsonConvert.SerializeObject(body);

return client.UploadString(Url + "\\" + id, "PUT", data);

}

public static string Delete<T>(T body, int id)

{

WebClient client = new WebClient();

client.Encoding = Encoding.UTF8;

client.Headers.Add(HttpRequestHeader.ContentType, "application/json");

string data = JsonConvert.SerializeObject(body);

string result = client.UploadString(Url + "\\" + id.ToString(), "DELETE", data);

return result;

}

}

# **4.2.2 Листинг интерфейса**

<ScrollView>

<StackLayout Padding="20"

Margin="0,10">

<Label Text="Программные продукты"

FontSize="28"

HorizontalOptions="Center"

Padding="0,5" />

<SearchBar x:Name="searchBar"

Placeholder="Введите название программного продукта"

TextChanged=" OnSearchProduct"

VerticalOptions="Start"

Margin="0,0,0,10"/>

<StackLayout Orientation="Horizontal"

HorizontalOptions="Center"

Spacing="10">

<Button Text="Добавить"

Clicked="btn\_Add\_Clicked"

FontSize="18"

Padding="10"/>

<Button Text="Редактировать"

Clicked="btn\_Edit\_Clicked"

FontSize="18"

Padding="10" />

<Button Text="Удалить"

Clicked="btn\_Delete\_Clicked"

FontSize="18"

Padding="10" />

</StackLayout>

<ScrollView>

<ListView x:Name="lv\_ Product"

HasUnevenRows="True">

<ListView.ItemTemplate>

<DataTemplate>

<ViewCell>

<Grid Padding="10">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="Auto" />

</Grid.ColumnDefinitions>

<StackLayout Grid.Column="0"

Padding="10">

<Label Text="{Binding Name}"

FontSize="20"

VerticalOptions="Center"

FontAttributes="Bold"/>

<Label Text="Версия: "

FontAttributes="Bold"

FontSize="15"

Padding="0,3"/>

<Label Text="{Binding Version}"

FontSize="15"

Padding="0,3"/>

<Label Text="Тип: "

FontSize="15"

FontAttributes="Bold"

Padding="0,3"/>

<Label Text="{Binding Type}"

FontSize="15"

Padding="0,3"/>

<Label Text="Фирма: "

FontAttributes="Bold"

FontSize="15"

Padding="0,3"/>

<Label Text="{Binding Firm}"

FontSize="15"

Padding="0,3"/>

<Label Text="Дата выпуска: "

FontAttributes="Bold"

FontSize="15"

Padding="0,3"/>

<Label Text="{Binding Date}"

FontSize="15"

Padding="0,3"/>

<Label Text="Прикладная область: "

FontAttributes="Bold"

FontSize="15"

Padding="0,3"/>

<Label Text="{Binding AppliedField}"

FontSize="15"

Padding="0,3"/>

<Label Text="Стоимость лицензии: "

FontAttributes="Bold"

FontSize="15"

Padding="0,3"/>

<Label Text="{Binding CostLicense}"

FontSize="15"

Padding="0,3"/>

</StackLayout>

</Grid>

</ViewCell>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</ScrollView>

</StackLayout>

# **5 Рефакторинг кода**

Код не требует никаких доработок.

# **ЗАКЛЮЧЕНИЕ**

Создание информационной системы для управления программными продуктами является важным шагом к оптимизации процессов управления программным обеспечением в организациях. В рамках этого проекта разработана структура и функциональные требования, направленные на автоматизацию ключевых задач, таких как учет лицензий, мониторинг инсталляций и анализ использования программных продуктов. Особое внимание уделено созданию интуитивно понятных интерфейсов, которые упростят работу ИТ-специалистов и позволят им более эффективно управлять ресурсами компании.

Процесс тестирования и внедрения системы подтвердил ее функциональность и способность интегрироваться в уже существующие рабочие процессы. Внедрение данной информационной системы значительно улучшит контроль за лицензиями, минимизирует затраты на приобретение и модификацию программного обеспечения, а также позволит быстро реагировать на изменения в потребностях бизнеса. Эти изменения, в свою очередь, повысят общую эффективность работы компании и улучшат качество обслуживания клиентов.

Таким образом, проект успешно выполнил все поставленные задачи и создал надежный фундамент для дальнейшего развития информационных систем в области управления программными продуктами. Этот подход не только повысит конкурентоспособность организации, но и улучшит ее репутацию в сфере технологий и инноваций.