

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ  
Заведующий кафедрой ЭВМ  
\_\_\_\_\_ Д.В. Топольский  
« \_\_\_ » \_\_\_\_\_ 2023 г.

Разработка менеджера лицензий для приложения работы с  
векторными анализаторами цепей  
ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ  
ЮУрГУ – 090301.2023.210 ПЗ ВКР

Руководитель работы,  
к.пед.н., доцент каф. ЭВМ  
\_\_\_\_\_ Ю.Г. Плаксина  
« \_\_\_ » \_\_\_\_\_ 2023 г.

Автор работы,  
студент группы КЭ-406  
\_\_\_\_\_ А.И. Денизбаева  
« \_\_\_ » \_\_\_\_\_ 2023 г.

Нормоконтролёр,  
ст. преподаватель каф. ЭВМ  
\_\_\_\_\_ С.В. Сяськов  
« \_\_\_ » \_\_\_\_\_ 2023 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ  
Заведующий кафедрой ЭВМ  
\_\_\_\_\_ Д.В. Топольский  
«\_\_» \_\_\_\_\_ 2023 г.

### **ЗАДАНИЕ**

**на выпускную квалификационную работу бакалавра**  
студенту группы КЭ-406  
Денизбаевой Альбине Искандеровне  
обучающемуся по направлению  
09.03.01 «Информатика и вычислительная техника»

- 1. Тема работы:** «Разработка менеджера лицензий для приложения работы с векторными анализаторами цепей» утверждена приказом по университету от «\_\_» \_\_\_\_\_ 20\_\_ г. № \_\_
- 2. Срок сдачи студентом законченной работы:** 1 июня 2023 г.
- 3. Исходные данные к работе:**

Требования к функционалу разрабатываемого приложения:

- а. Приложение Менеджер лицензий обеспечивает возможности:
- б. пользователю – активировать лицензию с использованием аппаратного или лицензионного ключей;
- с. владельцу защищаемого приложения SNVNA – управлять лицензированием программного обеспечения.
- д. Программа должна разрабатываться на языке программирования C++ с использованием фреймворка Qt.

**4. Перечень подлежащих разработке вопросов:**

1. Аналитический обзор научно-технической, нормативной и методической литературы по тематике работы.
2. Проектирование программной архитектуры.
3. Разработка программного решения.
4. Проведение тестирования программного решения.

**5. Дата выдачи задания:** 2 декабря 2022 г.

Руководитель работы \_\_\_\_\_ /Ю.Г. Плаксина/

Студент \_\_\_\_\_ /А.И. Денизбаева/

## КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Аналитический обзор научно-технической, нормативной и методической литературы по тематике работы	01.03.2023	
Проектирование программной архитектуры	01.04.2023	
Разработка программного решения	01.05.2023	
Проведение тестирования программного решения	19.05.2023	
Компоновка текста работы и сдача на нормоконтроль	24.05.2023	
Подготовка презентации и доклада	27.05.2023	

Руководитель работы \_\_\_\_\_ /Ю.Г. Плаксина/

Студент \_\_\_\_\_ /А.И. Денизбаева/

## АННОТАЦИЯ

Денизбаева А.И. Разработка менеджера лицензий для приложения работы с векторными анализаторами цепей — Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШ ЭКН; 2023, 64 с., 27 ил., библиогр. список — 22 наим.

В рамках выпускной квалификационной работы производится проектирование и разработка программного обеспечения – менеджер лицензий. Процесс выполнения выпускной квалификационной работы включает в себя решение следующих вопросов: обзор аналогов, обзор средств реализации, проектирование программной архитектуры, разработка программного решения, а также тестирования программного решения.

В работе рассмотрены существующие методы управления лицензиями, их преимущества и недостатки. Проведен анализ требований к менеджеру лицензий, определены функциональные возможности системы, разработаны архитектура и интерфейс пользователя.

Основная цель разрабатываемого приложения – избавить клиентов от привязки рабочих мест к приборам и повысить защиту программного обеспечения от несанкционированного копирования и распространения.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	7
1. АНАЛИТИЧЕСКИЙ ОБЗОР НАУЧНО-ТЕХНИЧЕСКОЙ, НОРМАТИВНОЙ И МЕТОДИЧЕСКОЙ ЛИТЕРАТУРЫ ПО ТЕМАТИКЕ РАБОТЫ.....	10
1.1. ОБЗОР АНАЛОГОВ .....	11
1.2. АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ .....	20
1.3. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ .....	29
1.4. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ.....	30
2. ПРОЕКТИРОВАНИЕ ПРОГРАММНОЙ АРХИТЕКТУРЫ .....	31
3. РАЗРАБОТКА ПРОГРАММНОГО РЕШЕНИЯ.....	34
4. ПРОВЕДЕНИЯ ТЕСТИРОВАНИЯ ПРОГРАММНОГО РЕШЕНИЯ .....	37
ЗАКЛЮЧЕНИЕ .....	48
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	49
ПРИЛОЖЕНИЯ.....	52

## ВВЕДЕНИЕ

Одна из ключевых задач в области радиотехники — анализ электрических цепей, для которого необходимо очень точное измерение. Для этой цели были созданы специальные устройства - анализаторы электрических цепей (рисунок 1). Они позволяют проводить точные измерения амплитудных характеристик системы, таких как коэффициент усиления, потери, коэффициент стоячей волны и т.д.

Компания ООО «Планар» [1] более 30 лет специализируется на разработке и производстве векторных анализаторов цепей.



Рисунок 1 — Четырехпортовый ВАЦ С2209 серии Кобальт производства ООО «Планар»

Для автоматизации измерений амплитудных характеристик СВЧ устройств ООО «Планар» было разработано программное обеспечение SNVNA. Данная разработка позволяет сократить время проведения измерения амплитудных характеристик системы и уменьшить нагрузку на операторов. Благодаря этому уменьшается риск погрешностей и искажения результатов проводимых измерений. В состав программного обеспечения SNVNA входят виртуальные версии

нескольких приборов, расширения для проведения математических вычислений и преобразований, инструменты для отображения данных о проведенных ранее измерениях в различных формах.

Чтобы получить право на использование копии программного обеспечения SNVNA пользователю необходимо пройти процесс лицензирования. На данный момент ООО «Планар» привязывает лицензию к серийному номеру анализатора. Другими словами, прибор является ключом при каждом запуске программного обеспечения. Векторный анализатор цепей является средством измерения и периодически его необходимо сдавать на поверку в метрологические службы согласно Федеральному закону № 102-ФЗ «Об обеспечении единства измерений» [2] и приказу 2510 от 31 июля 2020 г. «Об утверждении порядка проведения средств измерений, требования к знаку поверки и содержанию свидетельства о поверке» [3]. Оборудование может находиться в метрологических службах несколько месяцев и заказчики сталкиваются с такой проблемой, как невозможность использования программы SNVNA. Поэтому возникла потребность разработать приложение, которое устранил необходимость подключать прибор для работы в программе. Привязка лицензии к серийному номеру анализатора является неэффективной с точки зрения сохранности программного обеспечения, так как возможна подмена серийного номера или взлом SNVNA с помощью сторонних программ, что может повлечь убытки со стороны компании. Чтобы исключить данные проблемы, ООО «Планар» заинтересованы в разработке менеджера лицензий для своего программного обеспечения.

Менеджер лицензий [4-6] — это программное обеспечение, которое обеспечивает контроль и управление лицензиями на использование программного продукта. Оно позволяет организовать процесс лицензирования, а также отслеживать и управлять правами пользователей на доступ к программе. Менеджер лицензий может отслеживать использование лицензий, предоставлять информацию о доступных лицензиях и уведомлять о просрочке оплаты или



истечении срока действия лицензии. Он может использоваться как средство защиты авторских прав и обеспечения безопасности программного продукта.

Для достижения поставленной цели, необходимо решить следующие поставленные задачи:

1. Обзор аналогов с целью определения требований к функционалу.
2. Проектирование программной архитектуры.
3. Разработка программного решения.
4. Проведение тестирования программного решения.

## 1 АНАЛИТИЧЕСКИЙ ОБЗОР НАУЧНО-ТЕХНИЧЕСКОЙ, НОРМАТИВНОЙ И МЕТОДИЧЕСКОЙ ЛИТЕРАТУРЫ ПО ТЕМАТИКЕ РАБОТЫ

Современный мир становится все более цифровым, и использование программного обеспечения является неотъемлемой частью повседневной жизни. Вместе с тем, интеллектуальная собственность в IT-сфере ничем не отличается от интеллектуальной собственности в любой другой сфере. Каждый разработчик стремится защитить свое программное обеспечение с помощью авторского права.

Однако, среди способов нарушения авторских прав преобладают «пиратство», копирование и распространение материалов, а также плагиат. Пиратство подразумевает несанкционированное правообладателем использование программного обеспечения, которые защищены авторским правом. Согласно результатам исследования [7] в 2022 году «пиратским» контентом пользуются примерно 45%. Это подтверждает остроту проблемы нелегального контента.

Чтобы защитить свои программы от изменений или дополнений, таких как отключение от контроля лицензии, удаление функции проверки пароля и т.д. разработчики прибегают к инструментам для защиты кода. Лицензирование программного обеспечения помогает обеспечить неискаженность программы. Благодаря этому злоумышленники не могут украсть или взломать авторские алгоритмы путем исследования программного кода, а разработчики уверены в сохранности своих программ и могут успешно развивать бизнес, продавая свои разработки.

Менеджер лицензий — это важный инструмент для защиты авторских прав и управления лицензиями на программное обеспечение. Он позволяет разработчикам легко управлять лицензиями, следить за тем, как они использовались и кем, и сохранять целостность программного кода, что, в свою очередь, помогает более эффективно защищать бизнес и репутацию компании.

## 1.1 ОБЗОР АНАЛОГОВ

В данном аналитическом обзоре рассмотрены различные решения реализации лицензирования программного обеспечения. Для анализа были выбраны такие программы как Dr. Web, HASP License Manager и Autodesk Desktop Licensing Service.

### 1.1.1 Менеджер лицензий Dr.Web

Dr.Web [8] использует собственную систему лицензирования для своих продуктов. Приобретение лицензии Dr.Web происходит через официальные партнеров или путем скачивания установочного файла с официального сайта (рисунок 1.1) и последующей оплаты лицензии на сайте компании.

Система лицензирования Dr.Web имеет несколько типов лицензий:

- Персональная лицензия — покупается одним пользователем для нескольких компьютеров, используется только для некоммерческих целей.
- Корпоративная лицензия — покупается компанией и используется на всех компьютерах внутри организации.
- Сетевая лицензия — покупается организацией для использования на нескольких компьютерах в пределах одной локальной сети.

Кроме того, Dr.Web предлагает временное использование продукта с помощью пробных версий, а также возможность дополнительной защиты и улучшения функциональности с помощью опций и плагинов.

Для активации и управления лицензиями Dr.Web использует сервис управления лицензиями, который позволяет просматривать, обновлять и управлять лицензиями через интернет. Кроме того, Dr.Web предоставляет возможность использования мобильных приложений для управления и мониторинга состояния лицензий на удаленных устройствах.

**DR.WEB SECURITY SPACE (ЗАЩИТА ПК)**  
Максимальная свобода жизни в Сети без ограничений в безопасности

ВЫБЕРИТЕ

Срок лицензии	1 год
Количество ПК (1 серийный номер. Начинает действовать одновременно для всех устройств в лицензии после активации)	2 ПК
Скидка за Dr.Web-ки	<a href="#">Авторизуйтесь</a> , чтобы обменять Dr.Web-ки на скидку.
ПРОМОКОД	<input type="text"/>

Вы сможете защитить бесплатно такое же количество моб. устройств.

Если для продления лицензии Dr.Web Security Space куплена новая лицензия на отличающееся количество ПК, зарегистрировать данную лицензию в качестве продления будет невозможно.

ЭКОНОМИЯ (2): 790.00 Р  
СТОИМОСТЬ ЗАКАЗА: 1790 Р

[Купить](#)

- Надежная защита Dr.Web для Windows, macOS, Linux. Компоненты защиты: [Windows](#) | [macOS](#) | [Linux](#) | [Android](#)
- Защита для Android БЕСПЛАТНО! Используйте для активации Dr.Web для Android серийный номер/ключевой файл, полученный при покупке лицензии на Dr.Web.
- Сравните Dr.Web с продуктами [для бизнеса](#)
- [Лицензионное соглашение](#)

Рисунок 1.1 — Веб-сайт Dr.Web

Менеджер лицензий Dr.Web представляет собой диалоговое окно, в котором требуется ввести 16-тизначный серийный номер (рисунок 1.2). После пользователю будет доступна информация о действующей лицензии.

Однако в интернете можно найти достаточно много сайтов, на которых в открытом доступе хранятся ключи. Также, если была взломана электронная почта пользователя, есть риск, что злоумышленники перехватили письмо, в котором был указан серийный номер. Установить нелицензионный антивирус на мобильное устройство тоже не составляет особого труда, так как в интернете можно найти большое количество взломанных приложений Dr. Web.

**Менеджер лицензий**

Узнайте, какие продукты и компоненты защиты Dr.Web доступны в составе имеющейся у вас лицензии, включает ли ваша лицензия услуги технической поддержки (2), контактные данные поставщика лицензии. Информация о лицензии также доступна в вашем [личном кабинете](#).

**Ваш серийный номер Dr.Web**  
должен быть активированным

xxxx xxxx xxxx xxxx X

**E-mail, указанный при регистрации серийного номера**

Я даю согласие ООО «Доктор Веб» (правообладателю ПО Dr.Web) на обработку моих персональных данных.

**ОТПРАВИТЬ** >

Рисунок 1.2 — Менеджер лицензий Dr.Web

Особенности:

- имеется возможность покупки онлайн или через розничные сети;
- есть возможность выбора параметров лицензии;
- простая активация лицензии.

Недостатки:

- возможен взлом лицензии методом перебора серийных ключей;
- ключи активации для ПК-версии или взломанное приложение антивируса для мобильного устройства можно найти в интернете;
- для активации требуется интернет.

### 1.1.2 HASP License Manager — менеджер аппаратных лицензий для 1С

Аппаратные ключи [9] — это электронные устройства, используемые для защиты программных продуктов, таких как продукты 1С. Ключи могут быть использованы для управления и проверки лицензий на программное обеспечение, а также для защиты от несанкционированного доступа к программному обеспечению.

Аппаратные ключи для продуктов 1С называются HASP [10]. Данный ключ работает по принципу, что без его физического наличия программное обеспечение

не запустится. HASP предоставляет механизмы решения вопросов лицензирования программных продуктов 1С.

Процесс лицензирования через аппаратный ключ HASP [11] следующий:

1. Приобретение аппаратного ключа: покупатель приобретает аппаратный ключ как дополнительный товар к программным продуктам 1С.

2. Установка и подключение к ПК: после приобретения ключа его необходимо подключить к компьютеру, на котором устанавливается программное обеспечение 1С.

3. Активация лицензии: после подключения ключа происходит активация купленных лицензий на программное обеспечение 1С. Ключ распознается и устанавливает связь между защищенным программным обеспечением и ключом.

4. Контроль и мониторинг: после активации лицензии, управление происходит через HASP License Manager. Менеджер позволяет контролировать доступ пользователя к защищенному программному обеспечению 1С (рисунок 1.3).

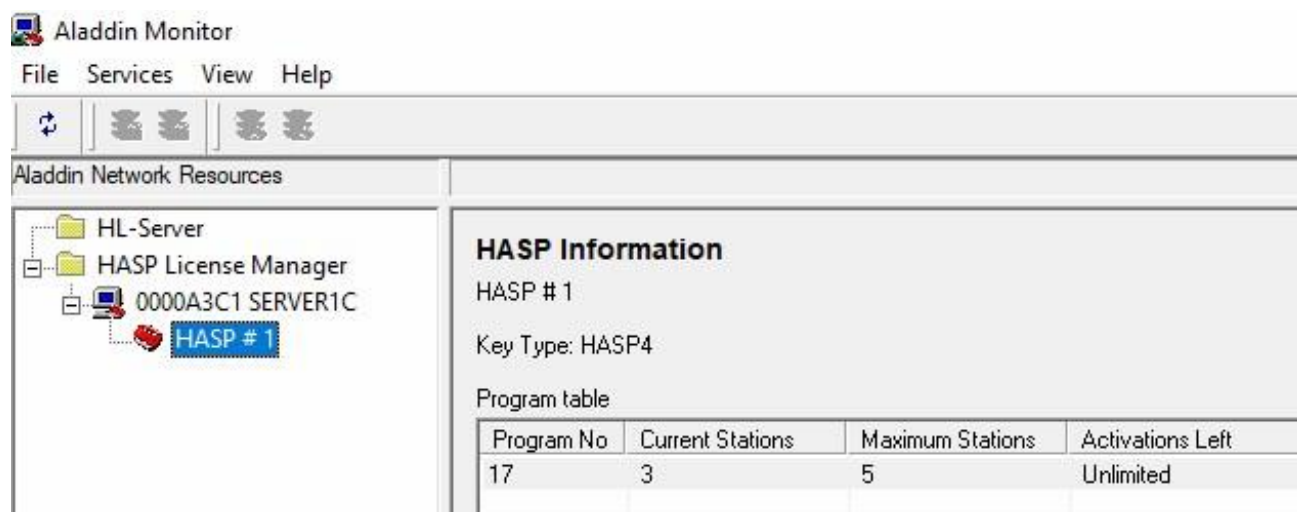


Рисунок 1.3 — Интерфейс HASP License Manager

Преимуществами такого метода лицензирования является то, что ключи HASP обеспечивают более надежную защиту лицензий на программное обеспечение 1С, так как ключи могут быть физически удалены и перенесены на другой компьютер, что упрощает перенос лицензий на другое оборудование. Также HASP License Manager обеспечивает удобный мониторинг и контроль за использованием

программного обеспечения, что повышает безопасность и эффективность его использования.

Для активации лицензии требуется интернет-соединение на компьютере пользователя. Если на компьютере не предусмотрен выход в сеть, то имеется возможность получить лицензию по телефону через оператора центра лицензирования, что является удобной альтернативой.

Особенности:

- согласно документации, на одну машину нельзя установить два hasp-ключа;
- высокая степень защиты;
- существуют альтернативные методы получения лицензии, если отсутствует интернет-соединение;

- ключ может предоставлять лицензии для нескольких пользователей.

Недостатки:

- потребуется время, чтобы аппаратный ключ был доставлен клиенту.

### 1.1.3 Autodesk Desktop Licensing Service

Autodesk Desktop Licensing Service [12] — это новый способ лицензирования программного обеспечения Autodesk, который был введен в 2017 году. Он используется для управления лицензиями Autodesk, включая AutoCAD, Revit, 3ds Max, Maya и др.

Для использования Autodesk Desktop Licensing Service необходимо установить приложение Autodesk Licensing Service на каждом компьютере, на котором будет использоваться программное обеспечение Autodesk. Это приложение связывает компьютер с сервером Autodesk, где хранятся все лицензии.

Процесс лицензирования через Autodesk Desktop Licensing Service следующий:

1. Приобретение лицензии: пользователь приобретает лицензию Autodesk и регистрируется на сайте Autodesk (рисунок 1.4).

2. Установка приложения Autodesk Licensing Service: после регистрации на сайте Autodesk пользователь устанавливает приложение Autodesk Licensing Service

на каждый компьютер, на котором будет использоваться программное обеспечение Autodesk.

3. Авторизация: по завершении установки Autodesk Licensing Service, пользователь авторизуется на сервере Autodesk, используя свой логин и пароль.

4. Получение лицензии: после авторизации, Autodesk Licensing Service получает лицензию от сервера Autodesk. Лицензия может быть временной или постоянной, в зависимости от типа приобретенной лицензии.

5. Использование программного обеспечения: после получения лицензии, пользователь может использовать программное обеспечение Autodesk на своем компьютере.

Преимуществами лицензирования через Autodesk Desktop Licensing Service является возможность легкого управления и мониторинга лицензий на все продукты Autodesk из одного места, а также возможность управления лицензиями в режиме онлайн, что облегчает перенос лицензий на другое оборудование. Кроме того, это обеспечивает удобство и улучшенную безопасность использования программного обеспечения Autodesk.

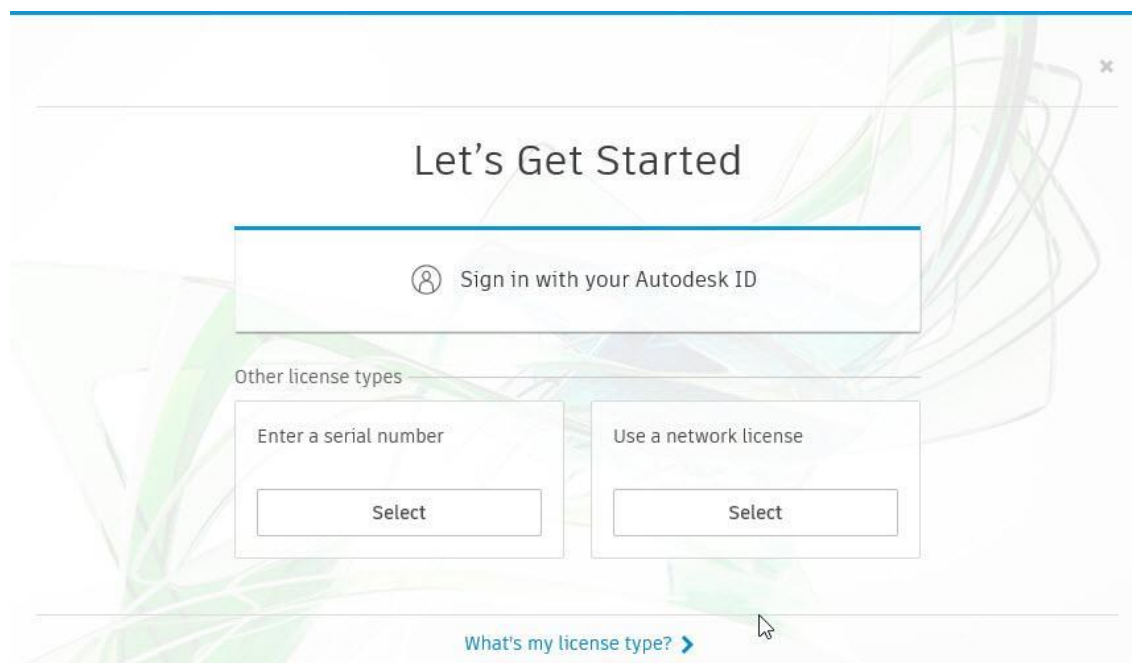


Рисунок 1.4 — Авторизация в Autodesk



Данный способ лицензирования программного обеспечения имеет ряд преимуществ, но также имеются весомые недостатки. В интернете можно найти множество способов взлома приложений Autodesk (рисунок 1.5). Также можно довольно быстро найти премиум версии приложений, которые не требуют наличия активированной лицензии.

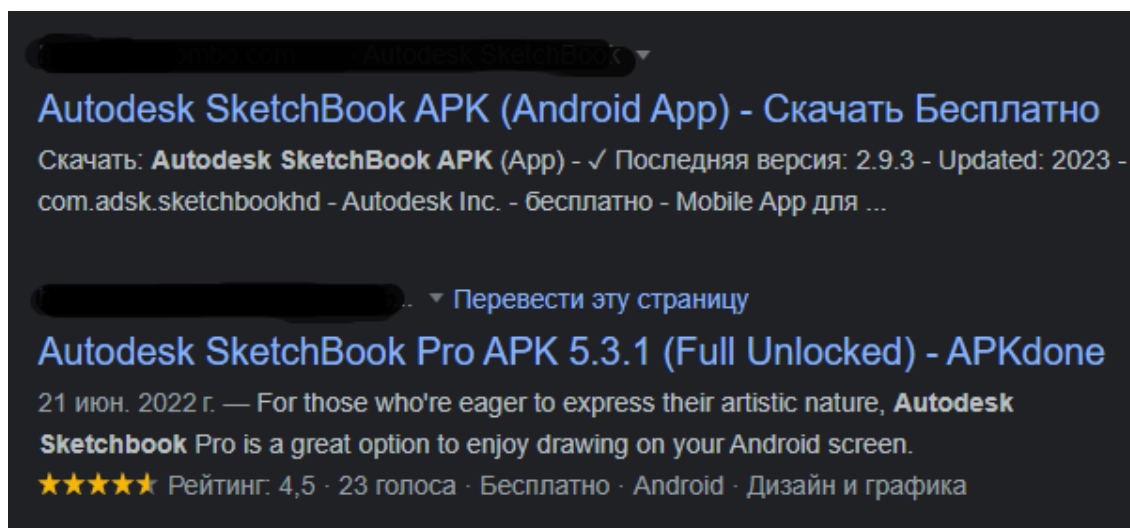


Рисунок 1.5 — Результаты по запросу «autodesk sketchbook apk»

Кроме того, если злоумышленники взломают электронную почту пользователя, есть риск навсегда потерять доступ к лицензии и восстановление потребует много времени и техподдержке потребуются доказательства, что вы действительно приобретали их программные продукты.

Особенности:

- нет необходимости в серийных ключах или аппаратных носителях даже в версиях, которые используются на больших предприятиях;
- можно быстро приобрести или продлить лицензию.

Недостатки:

- низкая степень защиты;
- необходимо постоянное интернет-соединение.

Основные преимущества и недостатки приведенных способов лицензирования и поставки программного обеспечения представлены в сравнительной таблице 1.

Таблица 1 — Сравнение аналогов

Метод лицензирования	Примеры использования	Преимущества	Недостатки
Dr.Web Licensing	Dr.Web for Windows, Dr.Web for Linux, Dr.Web for macOS, и др.	Множество опций лицензирования, гибкость настройки, возможность беспроводной интеграции в систему	Необходимость покупки стороннего шифровального программного обеспечения
HASP License Manager	Autodesk, Delcam, SolidWorks, PTC, и др.	Совместимость с различными операционными системами, возможность удаленной работы с лицензией	Дополнительные затраты на приобретение сервера, возможны проблемы с подключением и обновлением
Autodesk Desktop Licensing Service	Autodesk AutoCAD, Autodesk Revit, Autodesk Inventor, и др.	Упрощение процесса активации лицензии, управление лицензиями в централизованном режиме	Необходимость поддерживать соединение с сервером, высокая стоимость лицензирования и обслуживания

Таким образом, для компании, которая поставляет дорогостоящее программное обеспечение для бухгалтерского учета, документооборота или ведения бизнеса важна высокая защита продукта. Разработчики программного обеспечения, которое предназначено для широкой аудитории должны предоставить такой способ лицензирования, который позволит быстро распространить лицензии для всех пользователей и обеспечит полную конфиденциальность.

Разработчики рассмотренных в данном анализе программных решений защиты программного обеспечения позаботились о безопасности своих продуктов и о процессе восстановления лицензии в нештатных ситуациях.

Во всех рассмотренных аналогах недостатком является то, что для активации лицензии требуется подключение к интернету. Только 1С предоставляет второй способ активации лицензии — через оператора центра лицензирования.

Разрабатываемый менеджер лицензий направлен на защиту программного обеспечения для дорогостоящих приборов, предназначенных для анализа электрических цепей, которое ориентировано для специалистов определенной области. Поэтому разрабатываемый менеджер лицензий должен иметь все достоинства рассмотренных аналогичных программ.

## 1.2 АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ

### 1.2.1 Выбор среды разработки

Среда разработки Qt (Qt Creator) [13] — интегрированная среда разработки (IDE), разработанная для работы с Qt. Она предлагает различные функциональные возможности, такие как автодополнение кода, рефакторинг кода, графический редактор форм, связанный с исходным кодом, и многое другое.

Одним из наиболее привлекательных аспектов среды разработки Qt является поддержка кроссплатформенных приложений. Qt включает в себя набор библиотек, которые обеспечивают быстрый доступ к системным ресурсам и позволяют создавать кроссплатформенные приложения, где код можно запустить на разных операционных системах, таких как Windows, Mac, Linux и т. д., без каких-либо изменений в коде.

Среда разработки Qt отличается инновационной платформой для создания облачных приложений, благодаря интерфейсу интеграции с платформой Qt Cloud Services.

Другая важная функция Qt Creator, которая делает ее лучшим выбором для разработчиков — это отличная интеграция с версионными системами, такими как git, svn и более новыми, такими как Mercurial и Bazaar.

Для создания пользовательского интерфейса в Qt Creator есть визуальный редактор, который позволяет добавлять элементы интерфейса, такие как кнопки, поля ввода, список, таблицы и многое другое. Все элементы интерфейса могут быть настроены и спроектированы с помощью интуитивного и удобного в использовании пользовательского интерфейса.

Qt Creator хорошо подходит для разработки кроссплатформенных настольных приложений, мобильных приложений, веб-браузеров, 2D/3D приложений, игр и многих других приложений.

Инструкции по работе с средой разработки Qt [14] непрерывно обновляются и оптимизируются, что позволяет разработчикам быстро и эффективно создавать качественный код с использованием самых современных методик и технологий.

В целом, среда разработки Qt мощная и эффективная IDE, с множеством функций, удобной интеграцией и поддержкой многоплатформенной работы, делающих ее популярным выбором для многих разработчиков.

Среда разработки Visual Studio оптимизированная и универсальная IDE для разработки приложений на разных языках программирования, включая C++, C#, Java, Python и многие другие.

Одним из основных преимуществ Visual Studio является поддержка разных платформ, включая Windows, Android и iOS. Кроме того, эта IDE также обеспечивает поддержку взаимодействия с помощью GitHub и других систем управления версиями.

Visual Studio [15] имеет комфортный пользовательский интерфейс с простой навигацией. Среди технологик выделяются следующие:

1. Поддержка Live Share — новая функция, позволяющая разработчикам совместно работать и редактировать один и тот же файл в режиме реального времени.

2. Расширенный режим отладки — эта функция обеспечивает тщательный анализ кода, поддерживая выдачу ошибок и предупреждений на протяжении всего процесса создания приложения.

3. Интегрированный тестировщик — это позволяет разработчикам с легкостью внедрять и запускать автоматические тесты для проверки работоспособности кода.

4. Оптимизация производительности — с помощью ресурса Visual Studio Performance Monitor, разработчики могут отслеживать производительность, использование памяти и другие параметры, оптимизируя код для повышения производительности.

5. Расширяемость — Visual Studio позволяет дополнять функциональность, добавляя различные расширения для установки в среде разработки.

Visual Studio также обладает богатым и функциональным набором инструментов для разработки приложений, включая инструментарий разработки

веб-приложений ASP.NET, инструментарий разработки баз данных SQL Server и многие другие.

Помимо этого, Visual Studio — это платформа с простым и интуитивно понятным интерфейсом, которая обеспечивает программистам высокий уровень производительности, эффективности и надежности при работе с помощью использования мощной и гибкой среды для разработки.

Система разработки Qt и среда разработки Visual Studio представляют собой два разных инструмента для разработки приложений и имеют свои преимущества и недостатки. Вот некоторые из причин, по которым Qt может быть лучше Visual Studio:

1. Кроссплатформенность: Qt — это кроссплатформенная библиотека, которая позволяет разрабатывать приложения, работающие на разных ОС (Windows, macOS, Linux, Android, iOS и других). Использование Qt может существенно сократить временные затраты на адаптацию и перенос приложения между различными платформами.

2. Более широкий выбор языков программирования: Qt позволяет использовать не только язык C++, но и несколько других, таких как Python и JavaScript.

3. Наличие встроенных инструментов для разработки пользовательского интерфейса: Qt Creator — это среда разработки с отличной поддержкой для создания интерфейсов пользовательского приложения с использованием QT Widgets, QML и других технологий. Например, в Qt Creator есть встроенное графическое окружение, позволяющее разрабатывать пользовательский интерфейс, без необходимости вручную вводить код.

4. Поддержка множественных платформ, включая микроконтроллеры: Qt позволяет разработку не только на платформах с графическим интерфейсом, но и на микроконтроллерах и других встраиваемых системах.

5. Бесплатность и открытый исходный код: Qt может быть использован бесплатно и имеет открытый исходный код, что дает возможность для дополнительной настройки и оптимизации для конкретных нужд.

Qt Creator и Visual Studio — это две разные среды разработки с разными стилями и функциональными возможностями. Однако, Qt Creator лучше подходит для реализации менеджера лицензий по нескольким причинам.

Во-первых, Qt Creator напрямую опирается на библиотеку Qt, которая обладает мощными средствами для создания кроссплатформенных приложений. Это позволяет создавать приложения, работающие на Windows, Linux, macOS и других операционных системах без необходимости переписывания кода.

Во-вторых, Qt Creator имеет интуитивно понятный интерфейс и удобный редактор кода, облегчающие разработку и отладку кода. Благодаря интегрированному отладчику и поддержке профилирования приложений, Qt Creator делает разработку более эффективной и упрощает процесс отладки приложений.

В-третьих, Qt Designer — это интегрированный в Qt Creator инструмент для создания GUI приложений с помощью графического интерфейса. Это упрощает разработку приложений и позволяет сосредоточиться на бизнес-логике приложения, а не на деталях интерфейса.

Наконец, Qt Creator поддерживает C++, Python, JavaScript и другие языки программирования. Это означает, что разработчики могут выбирать наиболее подходящий язык программирования для своих задач.

Qt Creator обладает множеством преимуществ, которые делают его желаемой IDE для разработки программного обеспечения на C++ и Qt. Это надежный и интуитивно понятный инструмент, который позволяет быстро и эффективно создавать высококачественное программное обеспечение.

Сравнение сред разработки проводилось по ряду критериев, которые представлены в таблице 2.

Таблица 2 — Сравнение сред разработки по критериям

Характеристика	Visual Studio	Qt Framework
Языки программирования	C++, C#, F#, VB.NET, Python, HTML/CSS/JS	C++, QML, JavaScript, Python
Операционные системы поддержки	Windows	Windows, MacOS, Linux, iOS, Android
Интерфейс разработки приложений	Отдельная среда разработки (IDE)	Библиотека для разработки кросс-платформенных приложений (SDK)
Инструменты разработки	Отладчик, компилятор, профилировщик, автодополнение кода	Отладчик, компилятор, дизайнер на Drag & Drop, Qt Quick Designer
Кросс-платформенность	Ограничена только Windows	Полная кроссплатформенность
Разработка GUI и приложений	Полностью поддерживает разработку Native GUI на Windows и Xamarin Forms	Qt Creator поддерживает разработку GUI на всех поддерживаемых платформах

### 1.2.2 Выбор языка программирования

C++ — это статически типизированный язык программирования, который может быть использован для разработки высокопроизводительных приложений и системного программирования [16, 17]. Qt Creator написан на этом же языке, что позволяет использовать все функции Qt в приложениях на C++, соответственно, взаимодействие с интерфейсом будет происходить без задержек и преноса данных из Python-скриптов в C++ код. Кроме того, он обладает более низким уровнем абстракции, что дает возможность более тонкой настройки и оптимизации производительности, что может быть критично при работе с приложением/пакетом, связанным с обработкой лицензирования. Использование C++ и Qt позволит создать высокопроизводительное и надежное приложение с



кроссплатформенным графическим интерфейсом, которое легко интегрируется с базой данных клиентов. Для работы с базой данных в Qt можно использовать Qt SQL, который поддерживает работу с различными базами данных, такими как MySQL, PostgreSQL, SQLite и другими.

Кроме того, C++ является компилируемым языком, что означает, что код преобразуется в машинный код, который может быть непосредственно запущен на различных аппаратных платформах и для различных архитектур, что дает большую свободу для переносимости приложения.

Python — это высокоуровневый интерпретируемый язык программирования общего назначения, который создавался с акцентом на удобство и простоту использования [18]. Python обладает простым и чистым синтаксисом, который упрощает чтение и написание кода, и эффективной системой управления памятью, которая автоматически удаляет неиспользуемые объекты.

Одним из преимуществ Python является широкий спектр библиотек и модулей, которые доступны для различных задач, начиная от научных расчетов и машинного обучения до веб-разработки и автоматизации рутинных задач. Python также используется для разработки игр, научных исследований, системного администрирования и многих других задач.

В Python нет необходимости объявлять типы переменных, что упрощает написание кода и ускоряет его разработку. Python имеет широкое разнообразие стандартных функций, которые являются частью стандартной библиотеки. Также для Python доступно большое количество сторонних библиотек и модулей, которые делают язык еще более мощным и универсальным.

Python является одним из наиболее популярных языков программирования по сравнению с другими языками программирования, такими как Java, C++, Ruby и PHP. Это связано с тем, что Python легко осваивается, удобен и позволяет писать эффективный код в кратчайшие сроки.

При выборе языка программирования акцент ставился на такие важные критерии как метод обработки ошибок, скорость выполнения и возможность создания кроссплатформенных программ, которые отображены в таблице 3.

Таблица 3 — Сравнение языков программирования по критериям

Характеристика	C++	Python
Обработка ошибок	Сложнее и требует более глубоких знаний языка	Проще и более гибкий
Компиляция	Необходимо компилировать перед использованием	Можно использовать без предварительной компиляции
Производительность	Более высокая производительность за счет компиляции	Немного медленнее, но позволяет более быстро разрабатывать
Кроссплатформенность	Кроссплатформенность обеспечивается благодаря Qt	Кроссплатформенность обеспечивается благодаря Qt

Несмотря на то, что Python может быть полезным при работе с скриптами и автоматизации, C++ лучше подходит для более масштабных приложений, таких как менеджер лицензий. C++ имеет меньше встроенных библиотек, но они обеспечивают лучшую производительность разрабатываемых программ. Также, приложения, написанные на C++, меньше подвержены риску возникновения ошибок и поврежденных данных благодаря строгой обработке ошибок.

### 1.2.3 Выбор метода лицензирования

Лицензирование программного обеспечения [19] — это процесс наложения ограничений на использование программного обеспечения, установленных авторами и правообладателями. В случае программного обеспечения это означает разрешение на установку и использование программного обеспечения на конкретном компьютере.

Одним из наиболее популярных методов лицензирования программного обеспечения является использование лицензионных ключей [20]. Лицензионный ключ — это уникальный код, который лицензиар выдает лицензиату для использования его продукта. Зачастую введенный ключ должен быть активирован онлайн через сервер лицензиара.

Плюсы использования лицензионных ключей:

1. Лицензионные ключи легко использовать, и они часто сопровождаются подробными инструкциями по активации и обслуживанию.

2. Лицензионные ключи могут использоваться для разных видов программного обеспечения, что делает их полезными для организаций, работающих с различными продуктами.

3. Поскольку ключи могут быть предоставлены в электронном виде, их легко распространять по всему миру, а также быстро обновлять и активировать дополнительные пользовательские лицензии.

Минусы использования лицензионных ключей:

1. Лицензионные ключи могут быть взломаны, что приведет к несанкционированному использованию программного обеспечения.

2. Лицензионные ключи обычно ограничены по времени и числу использований.

3. Некоторые виды программного обеспечения не поддерживают использование лицензионных ключей, что может быть неудобным для пользователей.

Другим методом лицензирования программного обеспечения является использование защиты на основе аппаратных ключей [21], которые также называются «ключи защиты». Аппаратный ключ — это небольшое устройство, которое содержит информацию о лицензии. Эти ключи часто используются вместе с соответствующим программным обеспечением для защиты его от копирования и нелегального использования.

Плюсы использования аппаратных ключей:

1. Аппаратные ключи являются одним из наиболее надежных методов защиты программного обеспечения от нелегального использования.
2. Ключи защиты могут использоваться в разных устройствах и на разных операционных системах, делая их универсальными.
3. Ключи защиты легко подключаются к компьютеру и позволяют быстро защитить программное обеспечение от несанкционированного доступа.

Минусы использования аппаратных ключей:

1. Покупка и сопровождение аппаратных ключей может быть дополнительной финансовой нагрузкой для компании, особенно для малых и средних предприятий.
2. Некоторым пользователям может быть трудно использовать аппаратные ключи, особенно если они не имеют соответствующего технического опыта.
3. В случае потери или повреждения аппаратного ключа, может потребоваться дополнительная процедура восстановления, что может привести к дополнительным затратам времени и ресурсов.

### 1.3 Функциональные требования

#### Менеджер лицензий:

1. Выдача и управление лицензиями.
2. Мониторинг использования лицензий: функциональность должна предоставлять информацию по использованию выданных лицензий.
3. Шифрование и безопасность: менеджер должен обеспечивать безопасность данных и ключей, используя современные технологии шифрования. Также важно, чтобы была возможность отслеживания и регистрации инцидентов безопасности.
4. Оптимизация использования: менеджер лицензий должен предоставлять возможность оптимизировать использование лицензий, например, приостанавливая использование ключа при небольшой нагрузке.
5. Работа в автономном режиме: менеджер должен иметь возможность работать автономно, без постоянного подключения к интернету.
6. Соответствие со стандартами: менеджер должен соответствовать новейшим стандартам безопасности и конфиденциальности, а также следовать правилам, установленным в организации.
7. Расширяемость: менеджер должен иметь возможность управления лицензиями для большого числа программного обеспечения, а также расширяться для управления новыми продуктами и версиями.

#### Активатор лицензий:

1. Активация полученных клиентом лицензий на продукты компании.
2. Предоставление клиентам информации об активированных лицензиях.

#### 1.4 Нефункциональные требования

1. Программа должна быть разработана на языке C++ с использованием фреймворка Qt.
2. Все прикладное программное обеспечение системы должно поддерживать русский и английский языки.
3. Разрабатываемая система должна поддерживаться различных операционных системах.

## 2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОЙ АРХИТЕКТУРЫ

### 2.1 Основные сведения

Менеджер лицензий — это приложение, которое позволяет управлять лицензиями на программное обеспечение без подключения к Интернету. Некоторые функции, которые должны быть включены в менеджер лицензий, включают в себя:

1. Автоматический поиск установленных программных продуктов и связанных с ними лицензий на компьютере пользователя.

2. Отслеживание использования лицензий со стороны пользователей для обеспечения соблюдения пользовательского соглашения и контроля использования.

3. Генерация ключей активации лицензий, которые могут быть переданы пользователям, чтобы разрешить доступ к программному обеспечению.

4. Проверка действительности и принадлежности лицензий при попытке их использования, а также возможность блокировки лицензий при неисполнении условий пользовательского соглашения.

5. Возможность создания и отправки отчетов о использовании лицензий для администраторов программного обеспечения.

6. Обновление проверки лицензирования для поддержки новых версий программного обеспечения и актуальных стандартов безопасности.

7. Резервное копирование и восстановление параметров лицензирования, чтобы предотвратить потерю значимой информации.

8. Интерфейс управления пользователями и создание ролей для определения уровня доступа к функциям менеджера лицензий.

9. Возможность настройки различных параметров и настроек, таких как срок действия лицензий и лимиты использования.

Эти функции, в зависимости от конкретных потребностей программного обеспечения и пользовательских требований могут быть дополнены или изменены.

## 2.2. Проектирование структуры графического интерфейса

Интерфейс графического приложения версии для пользователя (рисунок 3.1)

состоит из одного основного окна:

- Информация о действующей лицензии.
- Кнопка "Зарегистрировать новую лицензию".
- Кнопка "Начать работу".

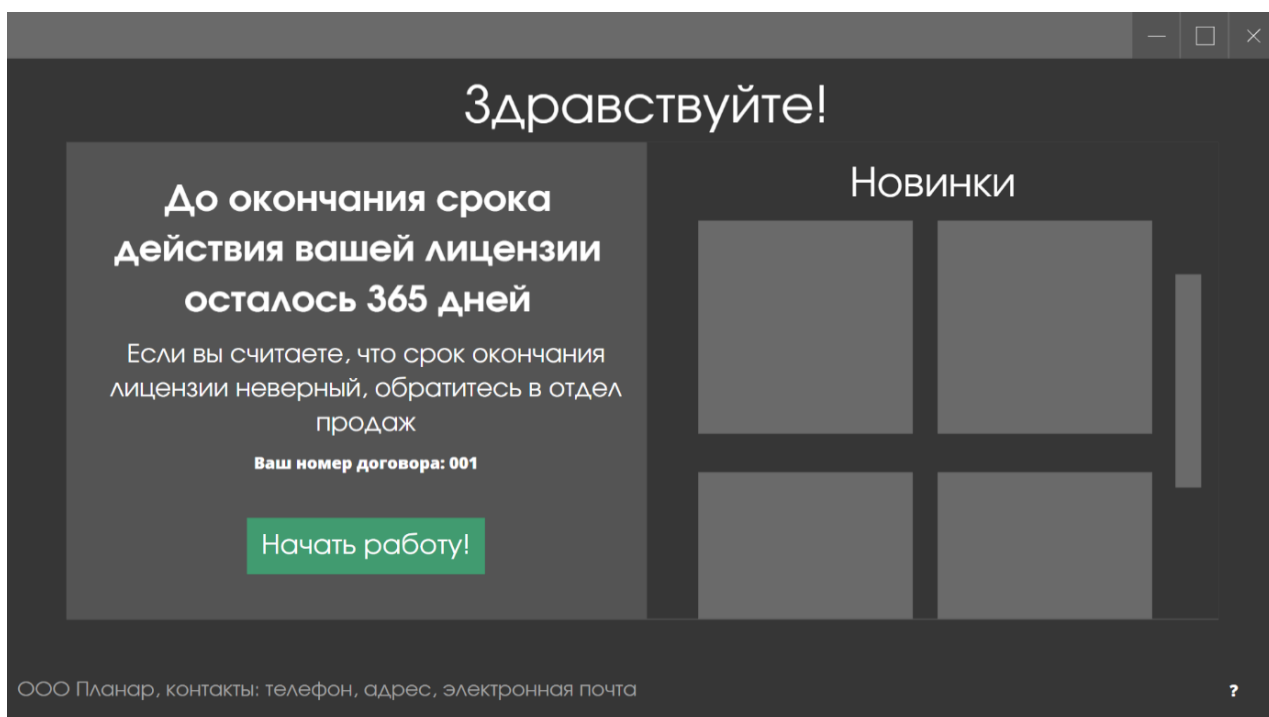


Рисунок 2.1 — Макет интерфейса активатора лицензии

После истечения срока действия лицензии кнопка в окне появится сообщение о необходимости активации новой лицензии, а кнопка "Запустить программу" станет неактивной. Для удобства пользователя в окне также будут отображены контакты отдела продаж поставщика программного обеспечения.



Интерфейс графического приложения версии для администратора (рисунок 3.2) состоит из двух окон: основного и диалогового. В основном окне приложения отображена таблица с информацией о выданных лицензиях и элементы управления лицензиями:

- Кнопка "Добавить лицензию".
- Кнопка "Изменить данные клиента".
- Кнопка "Удалить запись из таблицы".
- Таблица с информацией о выданных лицензиях.

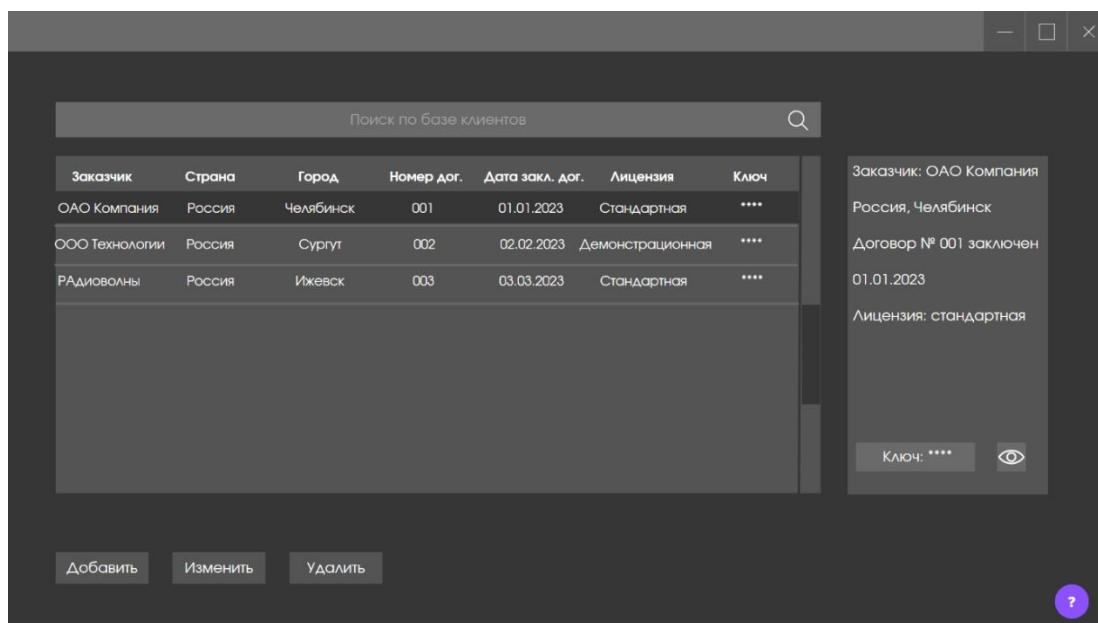


Рисунок 2.2 — Макет интерфейса менеджера лицензий

### 3 РАЗРАБОТКА ПРОГРАММНОГО РЕШЕНИЯ

При первом запуске менеджера лицензии для пользователя License Activation Master открывается окно, в котором необходимо ввести ключ активации. Для полной конфиденциальности ключ не отображается. После успешной активации, пользователю отображается вся необходимая ему информацией: дата активации лицензии, срок окончания, номер лицензии и контактные данные отдела продаж при возникновении вопросов (рисунок 3.1). По умолчанию программа запускается на русском языке. При нажатии на кнопку «Начать работу!» Запускается программное обеспечение SNVNA.

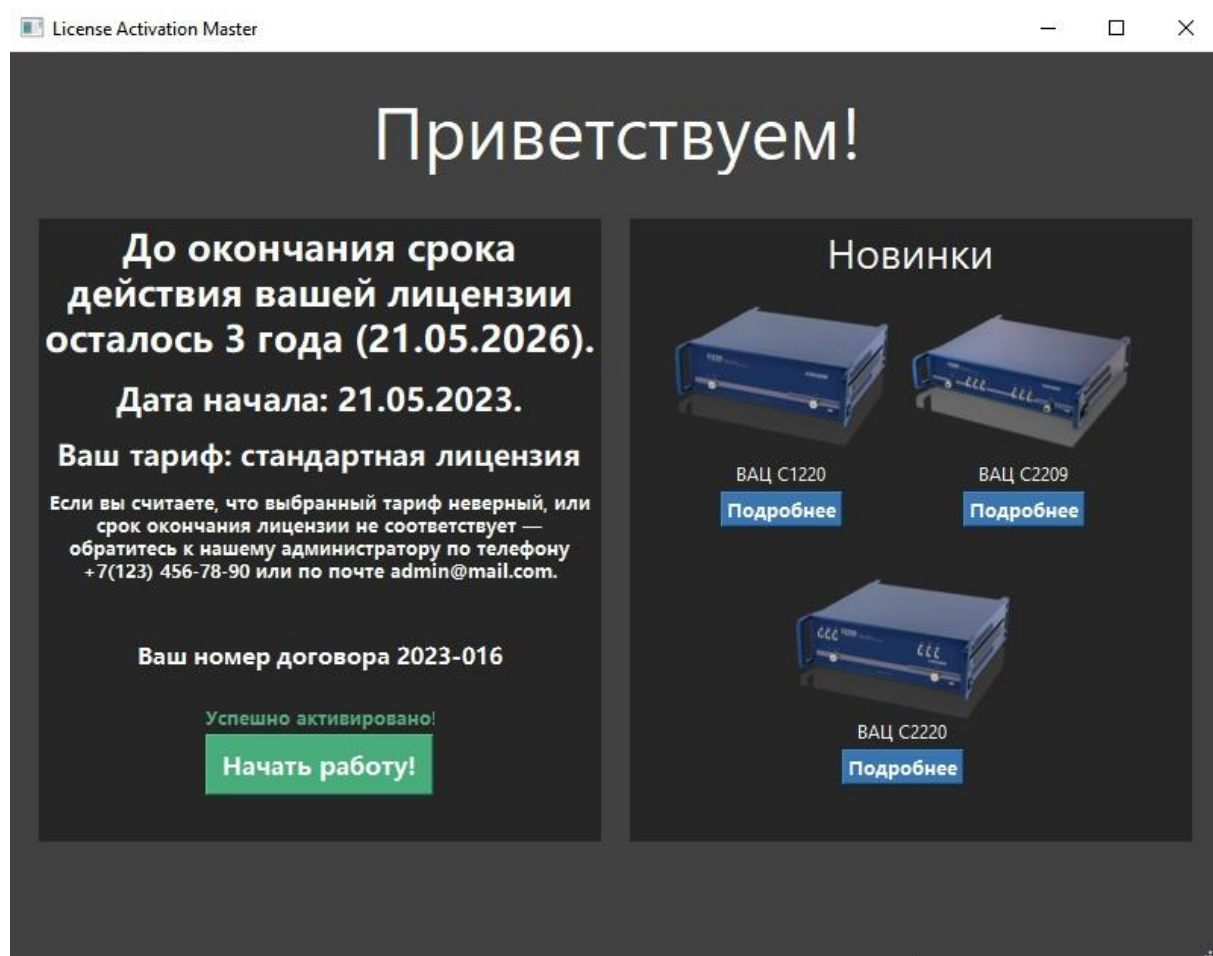


Рисунок 3.1 — Интерфейс программы License Activation Master

Программа для администратора License Manager отображает таблицу с информацией о всех выданных лицензиях (рисунок 3.2).

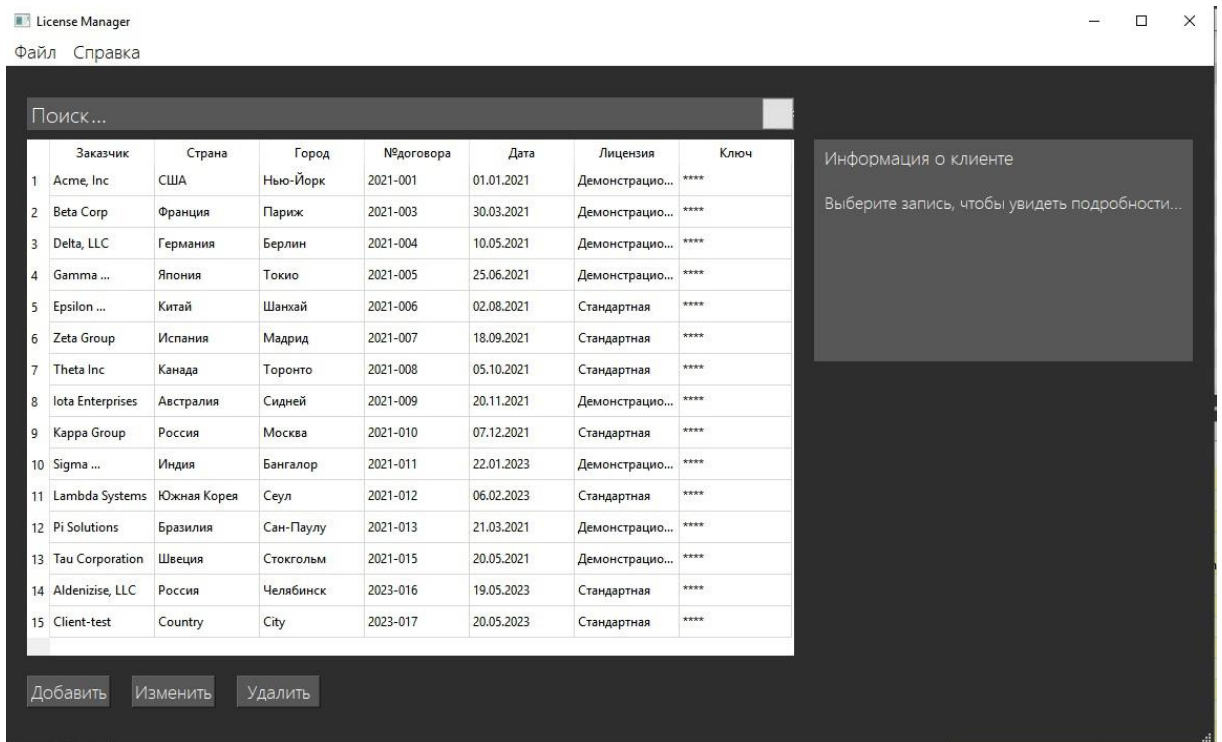


Рисунок 3.2 — Интерфейс программы License Manager

При нажатии на кнопку «Создать» открывается диалоговое окно для создания новой лицензии и полями для ввода информации о клиенте (рисунок 3.3).

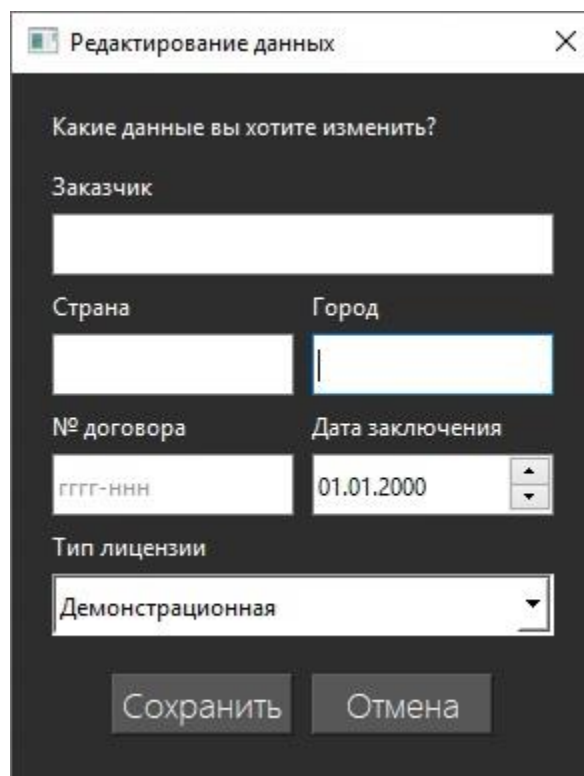


Рисунок 3.3 — Диалоговое окно редактирования данных

При нажатии на кнопку «Сохранить» данные вносятся в таблицу, кнопка «Отмена» закрывает диалоговое окно и удаляет введенные данные. В случае, если у клиента изменились какие-то данные их можно отредактировать, нажав на нужную строку и кнопку «Изменить». Запись из таблицы удаляется нажатием по нужной строке и кнопке «Удалить».

Разрабатываемая программа также должна защищать программное обеспечение. Для выполнения данной задачи будет использоваться метод с перемешиванием участков кода. Он заключается в том, что некоторые участки кода перемещаются в разные места или распределяются между разными файлами программного кода, после чего код перекомпилируется.

## 4 ПРОВЕДЕНИЯ ТЕСТИРОВАНИЯ ПРОГРАММНОГО РЕШЕНИЯ

### 4.1 Методология тестирования

Для тестирования программы был использован метод черного ящика [22], при котором тестирующий не знает внутренней структуры или реализации тестируемой программы. Он основан на проверке ввода/вывода и деятельности программы в различных ситуациях. В этом методе тестирующий сосредоточен на функциональности программы, исходя из ее спецификаций, требований и использования.

Метод черного ящика позволяет снизить зависимость от внутренней структуры приложения и сосредоточиться на том, как приложение взаимодействует с другими приложениями, как откликается на входящие данные и как выводит данные. Метод включает в себя множество тестирования, включая модульное тестирование, интеграционное тестирование, системное тестирование и приемочное тестирование.

### 4.2 Проведение процедуры тестирования

Тест-кейс на проверку корректности отображения элементов интерфейса:

1. Запустить программу.
2. Проверить, что все элементы интерфейса отображаются корректно: кнопки, меню, текстовые поля и т.д.

На рисунке 4.1 изображен интерфейс основного окна менеджера лицензий License Manager. В таблице отображается вся информация о всех заказчиках. Также в правой части можно увидеть подробную информацию о конкретном клиенте. Все элементы отображаются корректно.

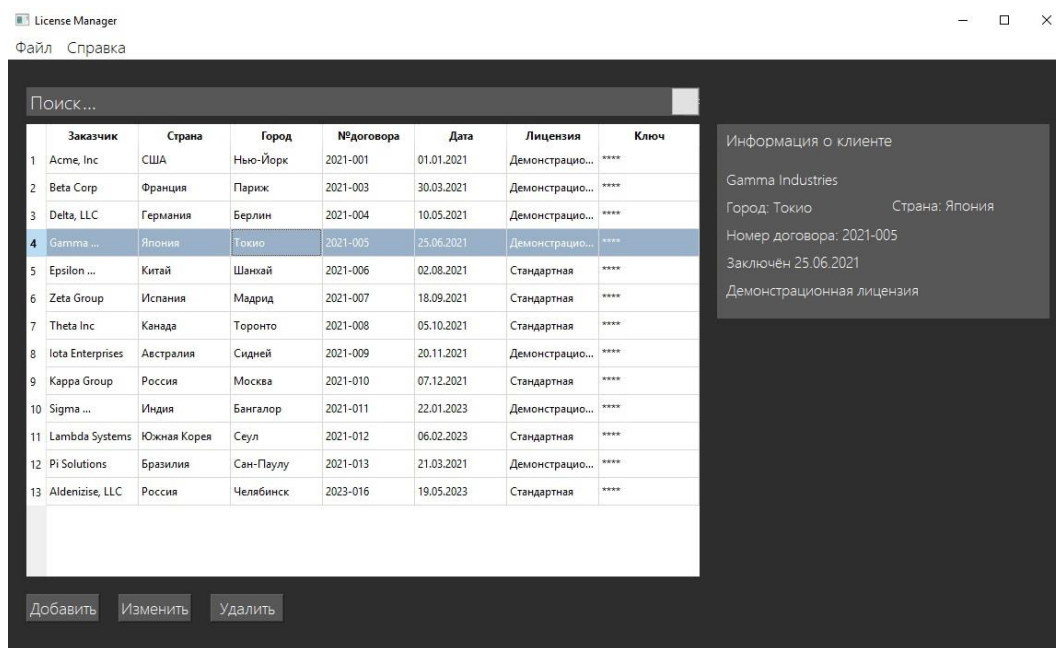


Рисунок 4.1 — Интерфейс основного окна License Manager

На рисунке 4.2 изображено диалоговое окно для создания новых записей и редактирования существующей информации. Все поля и кнопки отображаются корректно.

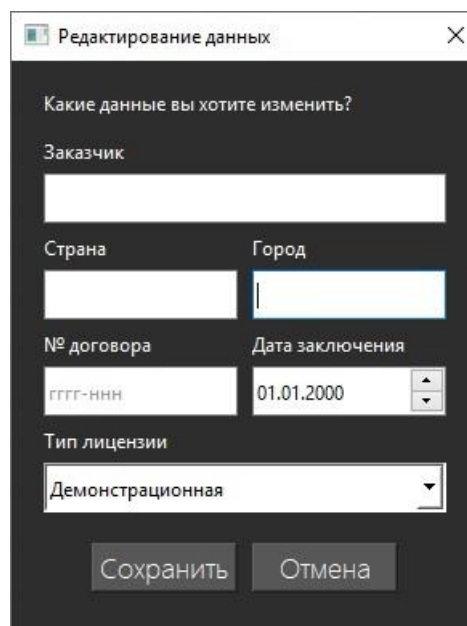


Рисунок 4.2 — Интерфейс диалогового окна License Manager

На рисунке 4.3 отображен интерфейс основного окна активатора лицензий License Activation Master. В левой части, при неактивной лицензии, отображается поле для ввода ключа, и кнопка «Активировать». В правой части можно увидеть новые анализаторы цепей производства «Планар».

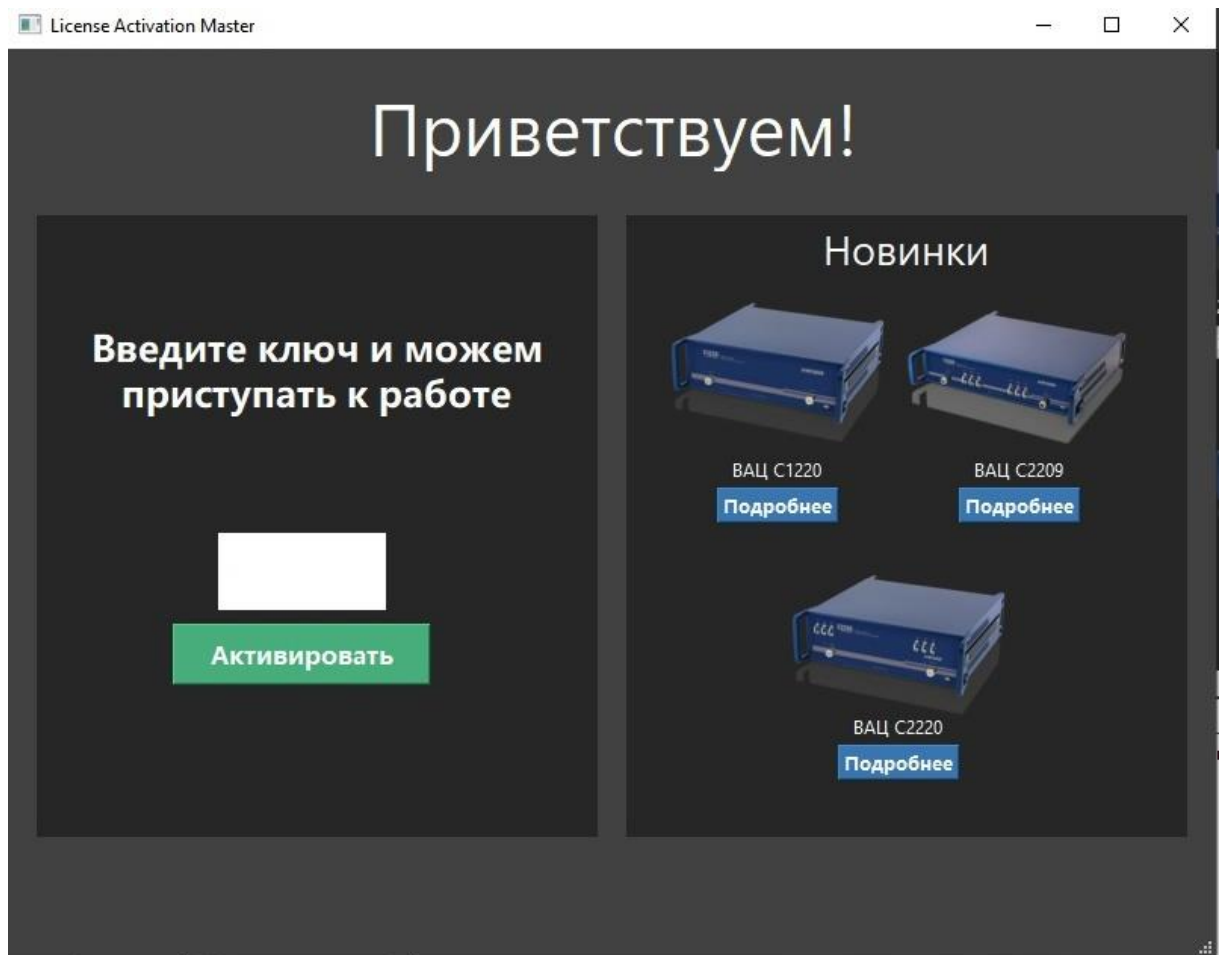


Рисунок 4.3 — Интерфейс основного окна License Activation Master

Тест-кейс на проверку корректности работы основных функций License Manager:

1. Запустить программу.
2. Создать новую лицензию.
3. Изменить данные в созданной лицензии.
4. Удалить лицензию.
5. Найти запись о лицензии по поиску

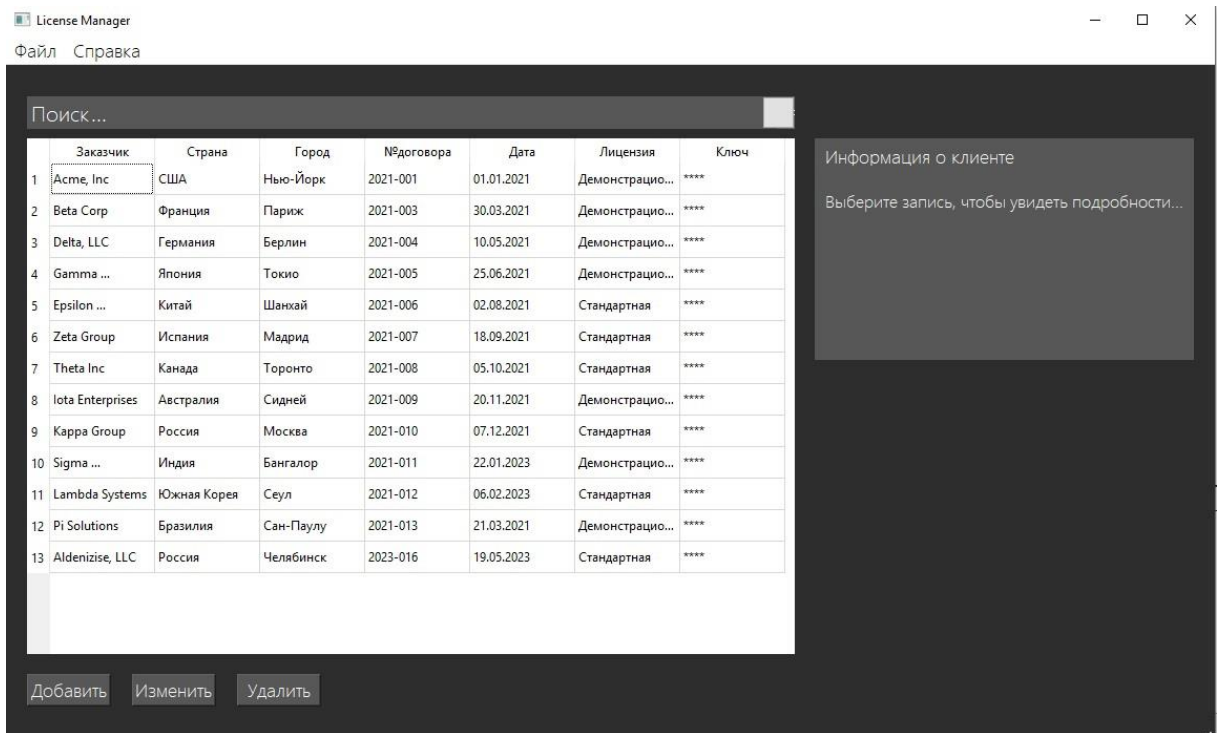


Рисунок 4.4 — Интерфейс основного окна License Manager

Чтобы создать новую лицензию, нужно нажать на кнопку «Добавить». Появится диалоговое окно для создания и редактирования данных (рисунок 4.5).

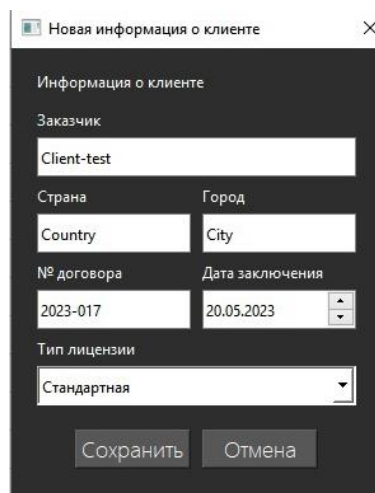


Рисунок 4.5 — Создание новой лицензии

При нажатии на кнопку «Сохранить» все введенные данные сохранятся в базе данных и будут отображены в таблице.



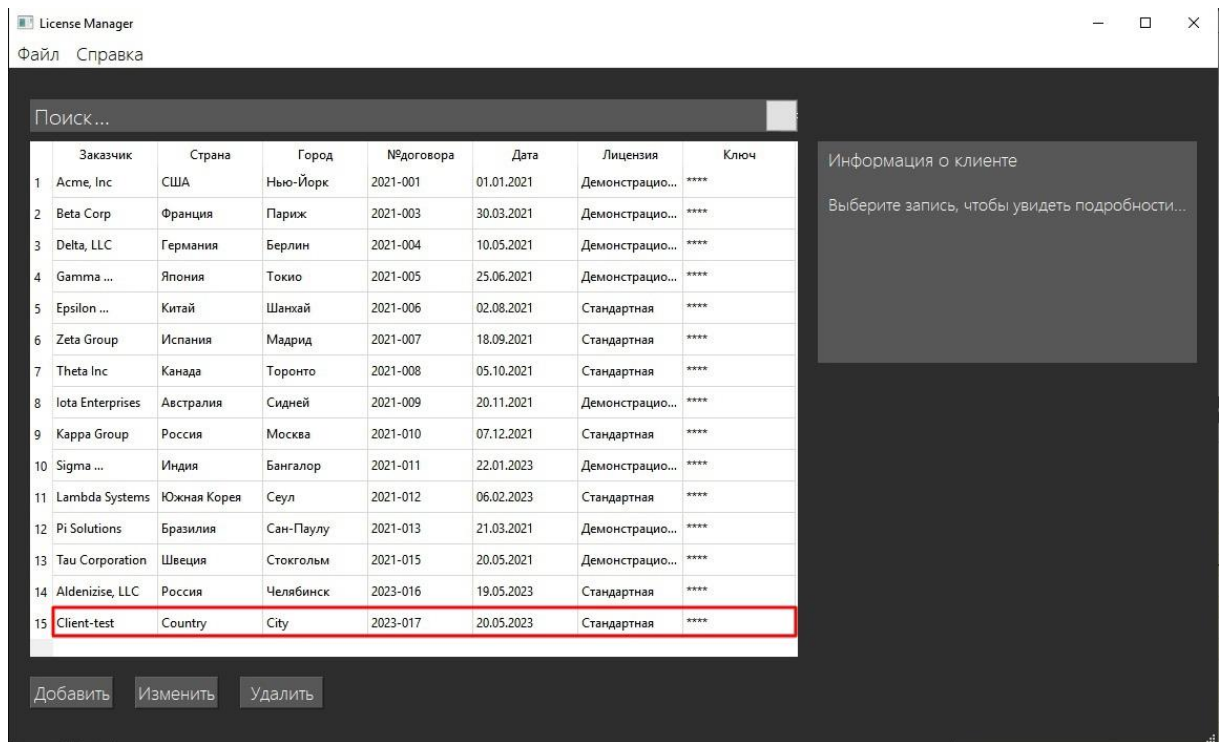


Рисунок 4.6 — Таблица с новой записью

Из сравнения рисунка 4.4 и рисунка 4.6 видно, что новая запись отображается внизу таблицы.

Чтобы отредактировать имеющиеся данные, необходимо выбрать нужную строку и нажать на кнопку изменить, откроется диалоговое окно с заполненными полями (рисунок 4.7). При нажатии на кнопку «Сохранить» все введенные данные обновятся в базе данных и также будут обновлены в таблице (рисунок 4.8).

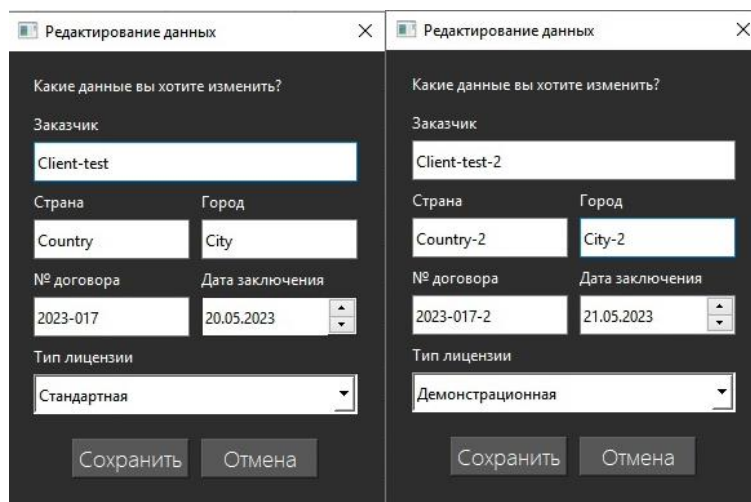


Рисунок 4.7 — Редактирование данных

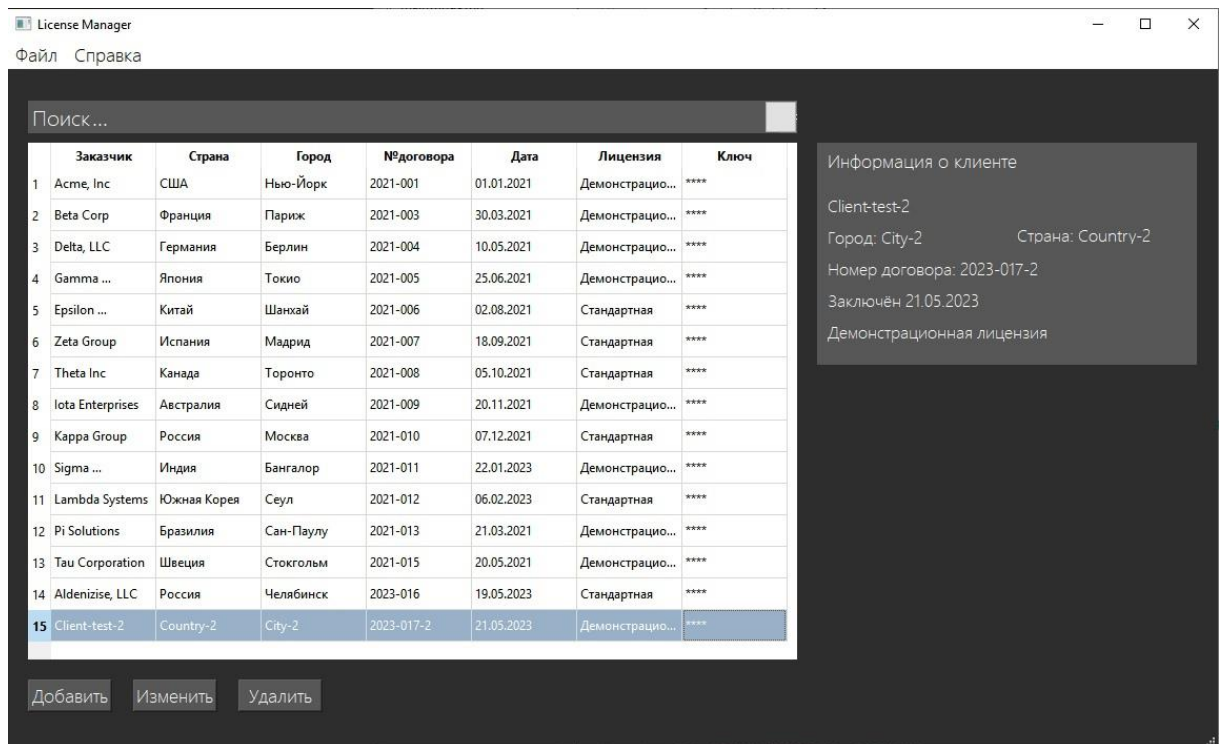


Рисунок 4.8 — Таблица с новой записью

Чтобы удалить лицензию, нужно выбрать запись и нажать на кнопку «Удалить». Появится информационное окно (рисунок 4.9), в котором нужно подтвердить удаление записи.

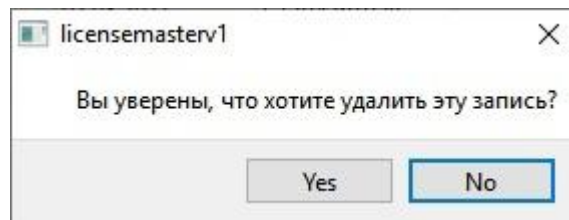


Рисунок 4.9 — Удаление записи

После подтверждения запись будет перенесена в архив и больше не будет отображаться в таблице.

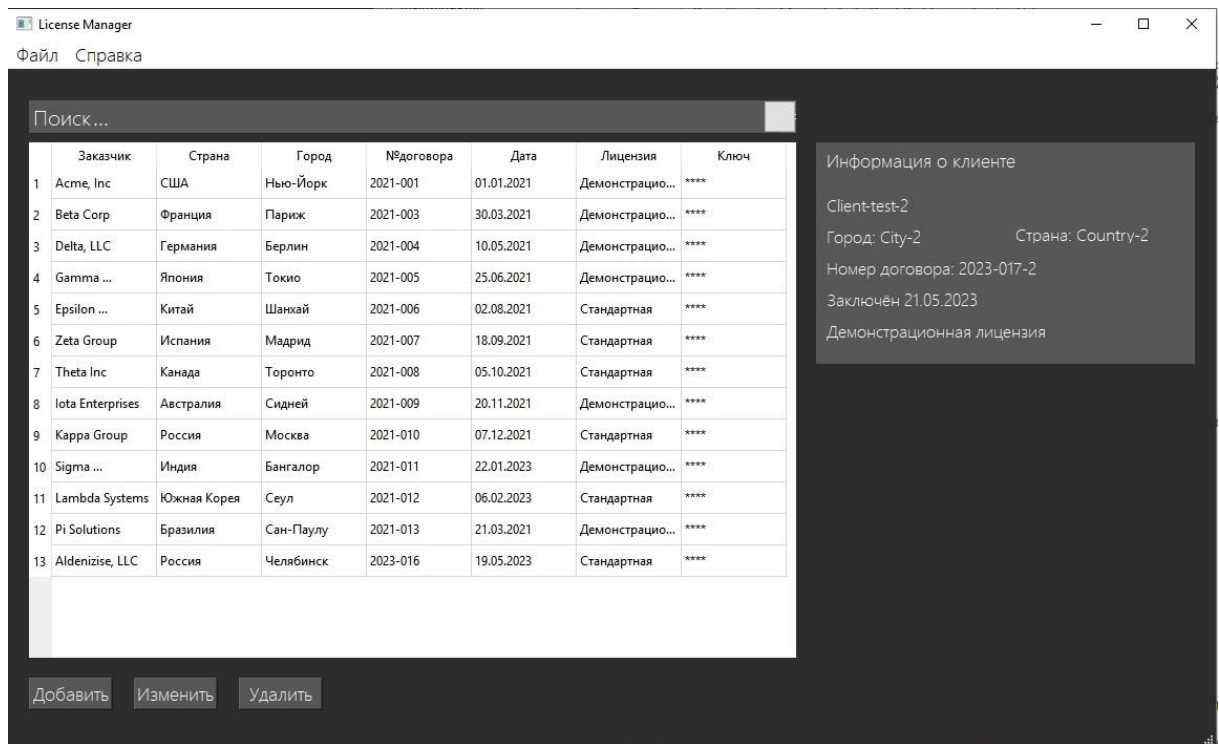


Рисунок 4.10 — Удаление записи

Для удобства пользователя в программе реализован поиск по всем параметрам. На рисунке 4.11 изображен результат работы программы при поиске существующей записи по запросу «Sigma».

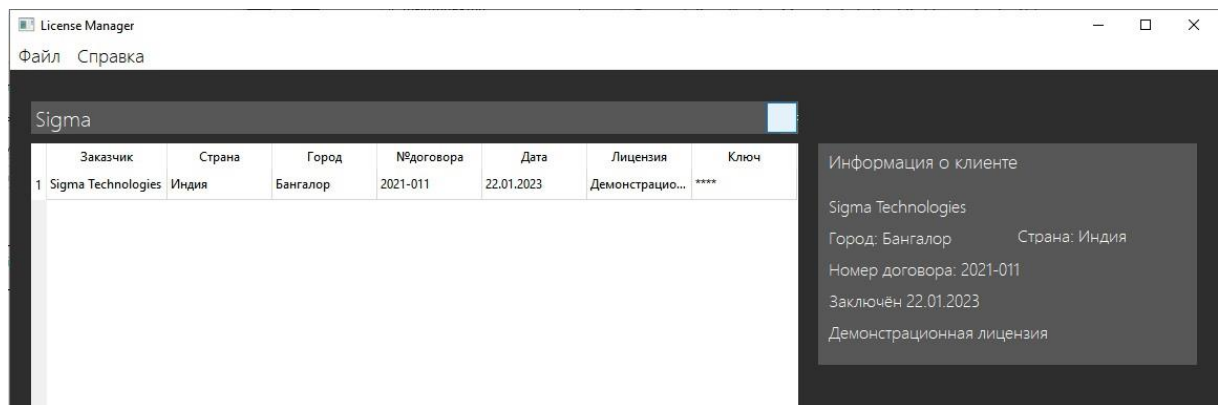


Рисунок 4.11 — Результат работы поиска по запросу «Sigma»

При попытке поиска несуществующей информации появится информирующее окно, что такой записи нет (рисунок 4.12).

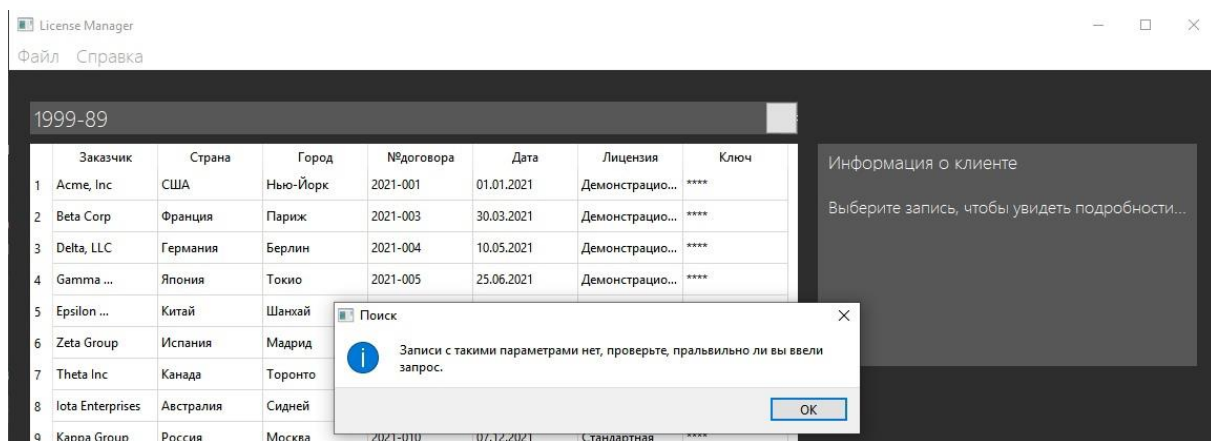


Рисунок 4.12 — Результат работы поиска несуществующей записи по запросу «1999-89»

Тест-кейс на проверку корректности работы основных функций License Activation Master:

1. Запустить программу.
2. Ввести некорректный ключ.
3. Ввести корректный ключ и активировать лицензию.

При первом запуске с неактивированной лицензией пользователю отображается основное окно License Activation Master с полем для ввода ключа (рисунок 4.13).

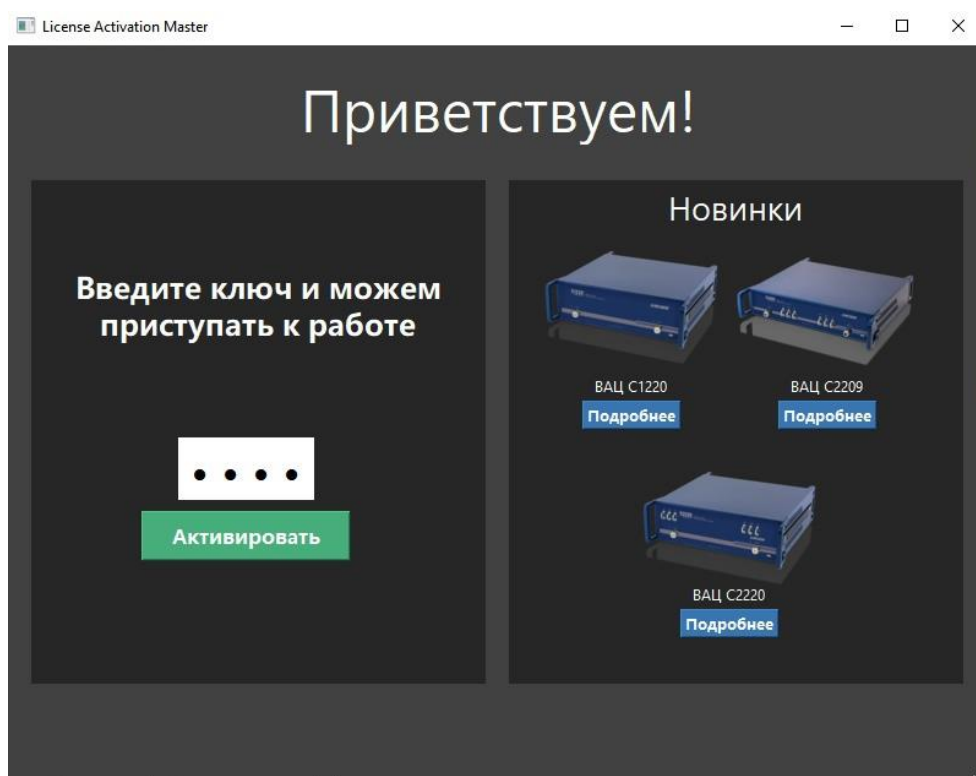


Рисунок 4.13 — Интерфейс основного окна License Activation Master

На рисунке 4.14 видно, что при вводе неверного ключа активировать лицензию невозможно. При возникновении такой ошибки настоящему пользователю будет предложено обратиться к администратору поставщика программного обеспечения для решения вопроса либо проверить ключ ещё раз.

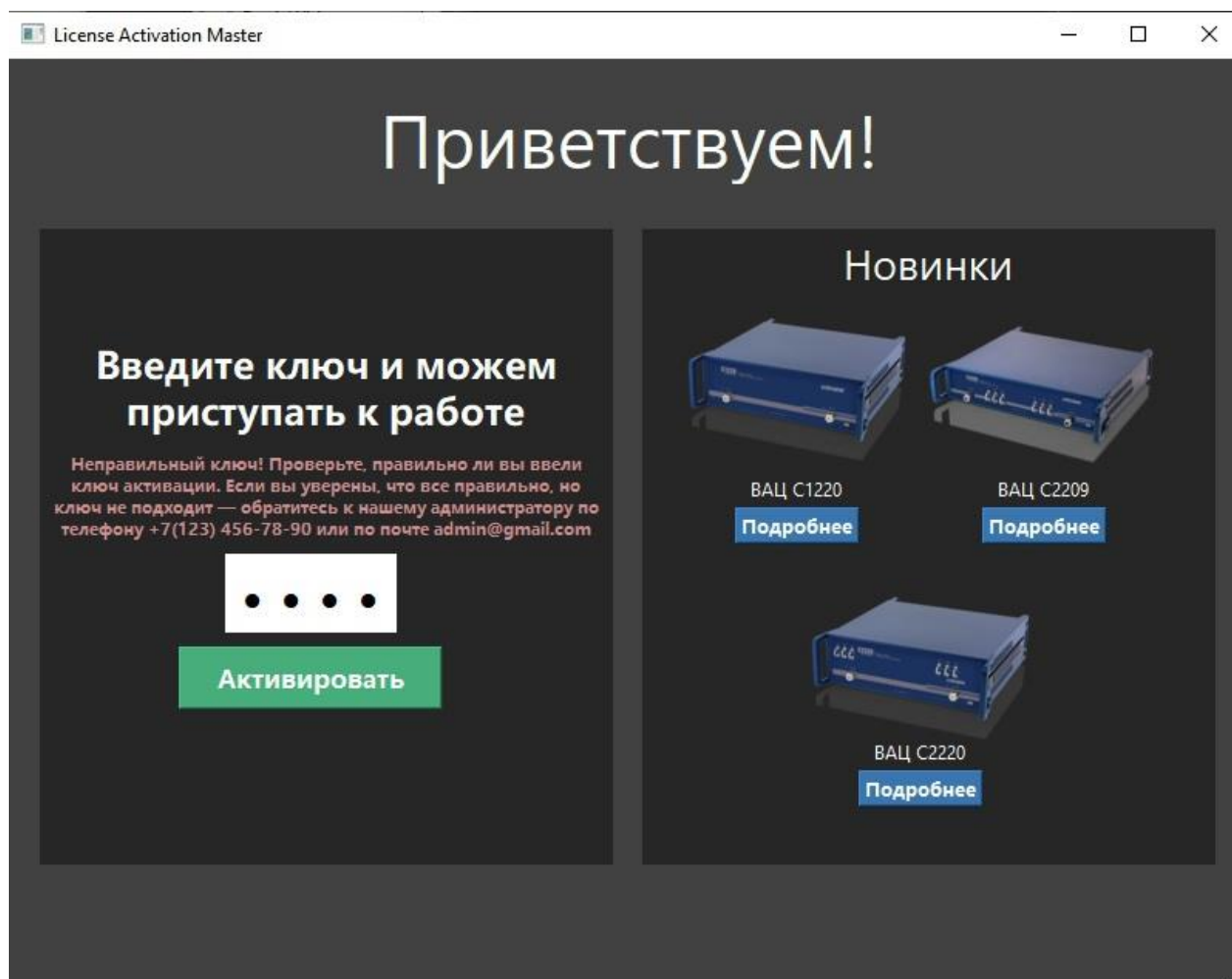


Рисунок 4.14 — Результат работы программы при вводе неверного ключа

Если ключ верный — лицензия становится активной и пользователь может увидеть дату начала лицензии и срок ее окончания, а также номер договора и контактные данные администратора поставщика программного обеспечения в случае возникновения таких вопросов, как неверный тариф, срок окончания лицензии, неправильный номер договора и т.д. (рисунок 4.15).

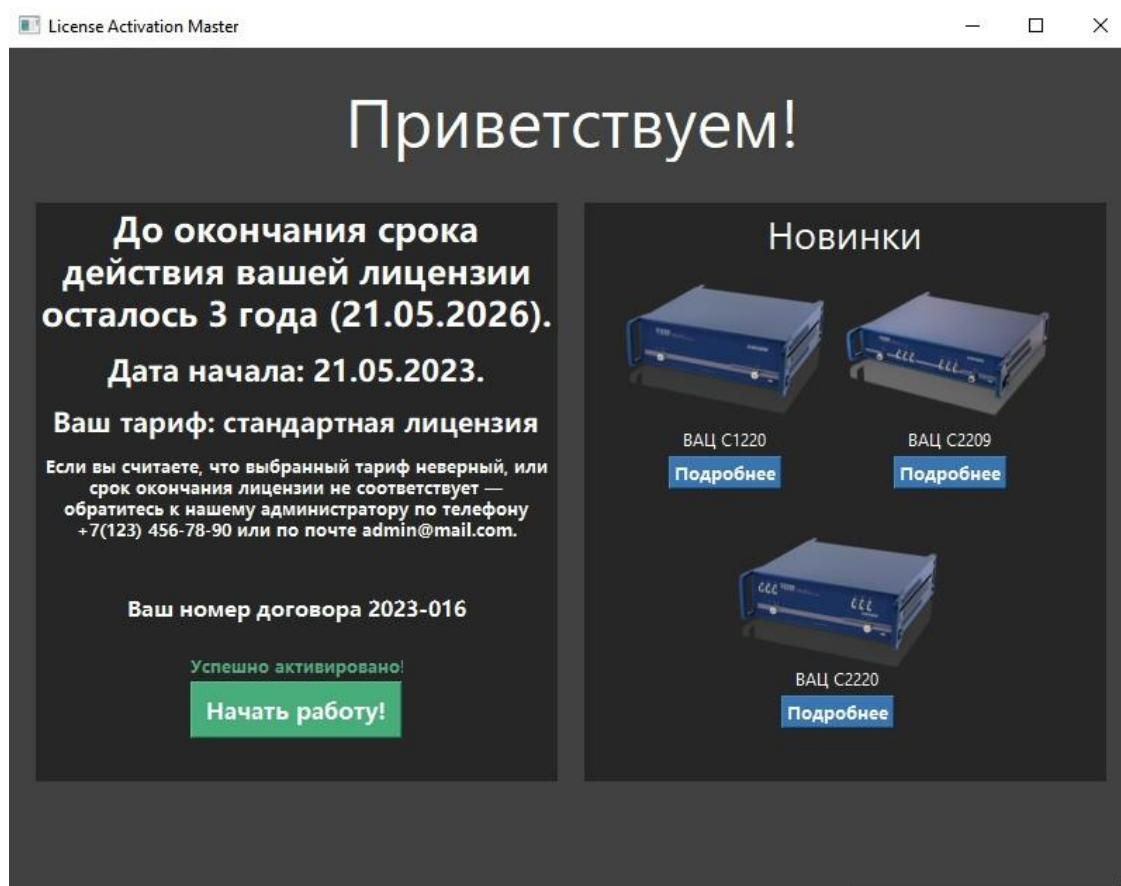


Рисунок 4.15 — Результат работы программы при вводе верного ключа  
Тест-кейс на проверку реакции программы на ошибки:

1. Запустить программу
2. Ввести специальные знаки в поле при создании новой лицензии или изменении данных.
3. Создать лицензию с пустыми данными.

Чтобы исключить ошибки при, при вводе данных в поля запрещено использовать специальные знаки, что видно на рисунке 4.16.

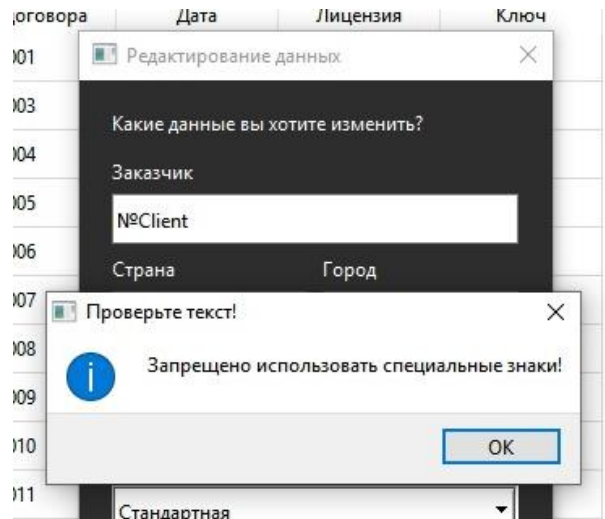


Рисунок 4.16 — Результат работы программы при введении запрещенных знаков

База данных не поддерживает пустые ячейки, поэтому создать новую лицензию, не заполнив основную информацию, невозможно (рисунок 4.17).

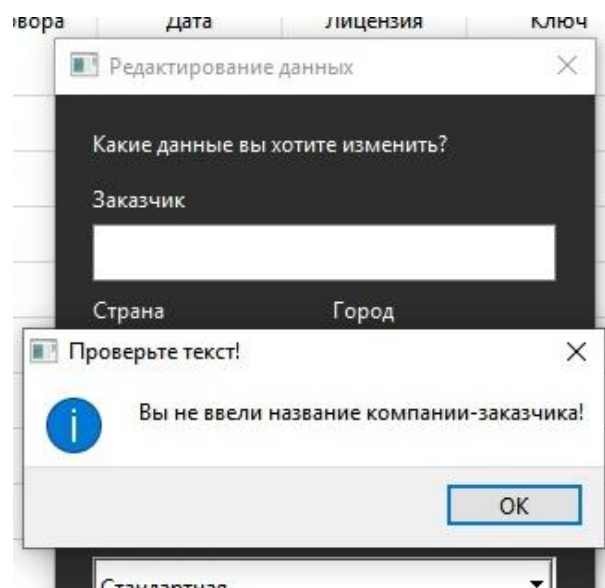


Рисунок 4.17 — Результат работы программы при незаполненном поле



## ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы была спроектированы и реализованы программы для управления лицензиями и их активации. Для достижения поставленной цели были выполнены следующие задачи:

1. Аналитический обзор научно-технической, нормативной и методической литературы по тематике работы.
2. Проектирование программной архитектуры.
3. Разработка программного решения.
4. Проведение тестирования программного решения.

В итоге можно отметить, что программа работает корректно. Реализованная программа поможет отслеживать выданные лицензии, кем и где используются и защитит программное обеспечение от взлома и несанкционированного копирования и распространения. В дальнейшем планируется сделать программу более устойчивой к хакерским атакам и реализовать дополнительный метод лицензирования, такой как использование аппаратных ключей. Так же планируется реализовать программу для других популярных платформ, таких как MacOS и Linux, чтобы программу можно было использовать на любом компьютере.



## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 ООО Планар [Электронный ресурс]. — URL: <https://planarchel.ru> (Дата обращения 1.03.2023).
- 2 Федеральный закон "Об обеспечении единства измерений" от 26 июня 2008 г. №102-ФЗ [Электронный ресурс]. — URL: <http://docs.cntd.ru/document/901875383> (Дата обращения 1.03.2023).
- 3 Приказ Минэкономразвития России от 31 июля 2020 г. №2510 "Об утверждении порядка проведения средств измерений, требования к знаку поверки и содержанию свидетельства о поверке". [Электронный ресурс]. — URL: <http://publication.pravo.gov.ru/Document/View/0001202008120025> (Дата обращения 1.03.2023).
- 4 Савельев, А. И. Лицензирование программного обеспечения в России: законодательство и практика : учебное пособие / А. И. Савельев. — Москва : Infotropic Media, 2012. — С. 431.
- 5 Гвоздева, Т. В. Проектирование информационных систем. Стандартизация, техническое документирование информационных систем : учебное пособие для спо / Т. В. Гвоздева, Б. А. Баллод. — 2-е изд., стер. — Санкт-Петербург : Лань, 2021. С. 214..
- 6 Корнилова, А. А. Защита персональных данных : учебное пособие / А. А. Корнилова, Д. С. Юнусова, А. С. Исмагилова. — Уфа : БашГУ, 2020. — С. 99.
- 7 Статья «Пираты 21 века: вернутся ли в России к нелегальному контенту» [Электронный ресурс]. URL: <https://www.sostav.ru/publication/piraty-xxi-veka-vernutsya-li-v-rossii-k-nelegalnomu-kontentu-54961.html>. (Дата обращения: 3.05.2023).
- 8 Официальный сайт Dr. Web [Электронный ресурс]. — URL: <https://www.drweb.ru>. (Дата обращения: 7.03.2023).
- 9 Электронный ключ, описание [Электронный ресурс]. — URL: [https://ru.wikipedia.org/wiki/Электронный\\_ключ](https://ru.wikipedia.org/wiki/Электронный_ключ) (Дата обращения: 7.03.2023).

- 10 "HASP License Manager", официальный сайт SafeNet [Электронный ресурс]. — URL: <https://sentinelcustomer.safenet-inc.com/sentineldownloads/> (Дата обращения: 7.03.2023).
- 11 "Техническое описание HASP", документация SafeNet [Электронный ресурс]. — URL: [https://docs.gemalto.com/hasp/hl/HASP\\_HL-TechOverview.pdf](https://docs.gemalto.com/hasp/hl/HASP_HL-TechOverview.pdf) (Дата обращения: 7.03.2023).
- 12 "Introducing Autodesk Desktop Licensing", официальный блог Autodesk [Электронный ресурс]. — URL: <https://blogs.autodesk.com/revit/2016/12/05/introducing-autodesk-desktop-licensing/> (Дата обращения: 7.03.2023).
- 13 Официальный сайт Qt [Электронный ресурс]. — URL: <https://www.qt.io/> (Дата обращения: 9.03.2023).
- 14 Документация Qt [Электронный ресурс]. — URL: <https://doc.qt.io> (Дата обращения: 9.03.2023).
- 15 Официальный сайт Visual Studio [Электронный ресурс]. — URL: <https://visualstudio.microsoft.com/ru/> (Дата обращения: 9.03.2023).
- 16 Пай, П. Реактивное программирование на C++ / П. Пай, П. Абрахам ; перевод с английского В. Ю. Винника. — Москва : ДМК Пресс, 2019. — С. 70.).
- 17 Мейерс Скотт "Эффективный и современный C++: 42 рекомендации по использованию C++11 и C++14", Издательство Вильямс 2019, с. 304.
- 18 Любанович Б. "Простой Python. Современный стиль программирования", 2-е издание, Издательство Питер, 2022, С. 592.
- 19 Александр Бакуткин, Вадим Губернаторов "Оптимизация стратегии продажи и лицензирования ПО", издательство "Омского государственного университета им. Ф. М. Достоевского", 2016, С. 47.
- 20 Дмитрий Пимонов "Лицензирование ПО с использованием ключей: преимущества, методы, стратегии", издательство "IT-Екатеринбург", 2016, С. 27.

- 21 Дюбов, А. С. Компьютерное обеспечение расчетно-проектной и экспериментально-исследовательской деятельности : учебное пособие / А. С. Дюбов. — Санкт-Петербург : СПбГУТ им. М.А. Бонч-Бруевича, 2021. — С. 34.
- 22 Старолетов, С. М. "Основы тестирования и верификации программного обеспечения" / С. М. Старолетов. — 3-е издание, стер. — Санкт-Петербург : Лань, 2023. — С. 14.

ПРИЛОЖЕНИЯ  
ПРИЛОЖЕНИЕ А

Исходный код основного окна License Manager

Листинг А1. mainwindow.h

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QSqlTableModel>

namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private slots:
    void on_deleteButton_clicked();
    void on_createButton_clicked();

private:
    Ui::MainWindow *ui;
    QSqlTableModel *model;
};
```

```
#endif // MAINWINDOW_H
```

Листинг А2. mainwindow.cpp:

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QMessageBox>
#include <QInputDialog>
#include <QSqlDatabase>
#include <QSqlQuery>
#include <QSqlError>
#include <QSqlRecord>

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);

    // Устанавливаем соединение с базой данных
    QSqlDatabase db = QSqlDatabase::addDatabase("QSQLITE");
    db.setDatabaseName("licenses.db");
    if (!db.open()) {
        QMessageBox::critical(this, "Ошибка", "Не удалось открыть базу данных");
        return;
    }

    // Создаем модель для таблицы
    model = new QSqlTableModel(this, db);
    model->setTable("licenses");
    model->setEditStrategy(QSqlTableModel::OnManualSubmit);
```

```
model->select();

// Устанавливаем заголовки для столбцов
model->setHeaderData(0, Qt::Horizontal, "ID");
model->setHeaderData(1, Qt::Horizontal, "Клиент");
model->setHeaderData(2, Qt::Horizontal, "Лицензия");

// Устанавливаем модель для таблицы
ui->tableView->setModel(model);
ui->tableView->setSelectionBehavior(QAbstractItemView::SelectRows);
ui->tableView->setSelectionMode(QAbstractItemView::SingleSelection);
ui->tableView->resizeColumnsToContents();
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::on_deleteButton_clicked()
{
    // Получаем выбранную строку
    QModelIndex index = ui->tableView->currentIndex();
    if (!index.isValid()) {
        QMessageBox::warning(this, "Ошибка", "Не выбрана строка для удаления");
        return;
    }

    // Проверяем пароль администратора
```

```
QString password = QDialog::getText(this, "Подтверждение", "Введите
пароль администратора", QLineEdit::Password);
    if (password != "admin") {
        QMessageBox::warning(this, "Ошибка", "Неверный пароль
администратора");
        return;
    }

    // Удаляем запись из таблицы и базы данных
    int row = index.row();
    model->removeRow(row);
    if (!model->submitAll()) {
        QMessageBox::critical(this, "Ошибка", "Не удалось удалить запись из базы
данных");
        model->revertAll();
        return;
    }
}

void MainWindow::on_createButton_clicked()
{
    // Создаем диалоговое окно для ввода информации о клиенте
    QDialog dialog(this);
    dialog.setWindowTitle("Создание записи");

    QFormLayout form(&dialog);

    QLineEdit clientEdit;
    form.addRow("Клиент:", &clientEdit);
```

```
QLineEdit licenseEdit;
form.addRow("Лицензия:", &licenseEdit);

QDialogButtonBox buttonBox(QDialogButtonBox::Ok | QDialogButtonBox::Cancel,
Qt::Horizontal, &dialog);
form.addRow(&buttonBox);

// Обрабатываем нажатие кнопки "Ок"
connect(&buttonBox, &QDialogButtonBox::accepted, &dialog, &QDialog::accept);
connect(&buttonBox, &QDialogButtonBox::rejected, &dialog, &QDialog::reject);

if (dialog.exec() != QDialog::Accepted) {
    return;
}

// Добавляем запись в таблицу и базу данных
QSqlRecord record = model->record();
record.setValue("client", clientEdit.text());
record.setValue("license", licenseEdit.text());
if (!model->insertRecord(-1, record)) {
    QMessageBox::critical(this, "Ошибка", "Не удалось добавить запись в базу
данных");
    model->revertAll();
    return;
}

// Генерируем ключ для активации лицензии
QString key = QString("%1-%2").arg(clientEdit.text()).arg(licenseEdit.text().length());
```



## Окончание приложения А

```
QMessageBox::information(this, "Создание записи", QString("Запись успешно  
создана\nКлюч для активации лицензии: %1").arg(key));  
}
```

## ПРИЛОЖЕНИЕ Б

### Исходный код основного окна License Activation Master

Листинг Б1. mainwindow.cpp.

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);

    setWindowTitle("Активатор лицензий");

    m_widget = new QWidget(this);
    setCentralWidget(m_widget);

    m_label = new QLabel("Товары компании С", m_widget);
    m_label->setAlignment(Qt::AlignCenter);

    m_productsWidget = new QWidget(m_widget);
    m_productsLabel = new QLabel("Список товаров", m_productsWidget);

    m_layout = new QStackedLayout(m_widget);
    m_layout->addWidget(m_productsWidget);
    m_layout->addWidget(m_activateWidget);

    m_activateWidget = new QWidget(m_widget);
    m_activateBtn = new QPushButton("Ввести ключ", m_activateWidget);
    m_keyLineEdit = new QLineEdit(m_activateWidget);
    m_licenseLabel = new QLabel(m_activateWidget);
```

```
m_okBtn = new QPushButton("Активировать", m_activateWidget);

m_infoLabel = new QLabel(m_widget);
m_infoLabel->setText("Если вы считаете, что выбранный тариф неверный, или
срок окончания лицензии не соответствует — обратитесь к нашему администратору по
телефону +7(123) 456-78-90 или по почте admin@mail.com. Ваш номер договора --");
m_contactLabel = new QLabel(m_widget);

m_snvnaBtn = new QPushButton("Запустить SNVNA", m_widget);
m_snvnaBtn->setDisabled(true);

QVBoxLayout *productsLayout = new QVBoxLayout(m_productsWidget);
productsLayout->setAlignment(Qt::AlignHCenter);
productsLayout->addWidget(m_productsLabel);

QVBoxLayout *activateLayout = new QVBoxLayout(m_activateWidget);
activateLayout->addWidget(m_keyLineEdit);
activateLayout->addWidget(m_licenseLabel);
activateLayout->addWidget(m_okBtn);

QVBoxLayout *contactLayout = new QVBoxLayout;
contactLayout->addWidget(m_infoLabel);
contactLayout->addWidget(m_contactLabel);
contactLayout->addWidget(m_snvnaBtn);
QHBoxLayout *mainLayout = new QHBoxLayout(m_widget);
mainLayout->addWidget(m_label);
mainLayout->addLayout(m_layout);
mainLayout->addLayout(contactLayout);

m_db = QSqlDatabase::addDatabase("SQLITE", "myconnection");
```

```

m_db.setDatabaseName("customersbase.db");
if(!m_db.open())
    QMessageBox::warning(this, "Ошибка", "Невозможно открыть базу данных: "+
m_db.lastError().text());
else
    qDebug() << "База данных открыта успешно";

connect(m_activateBtn, SIGNAL(clicked()), m_layout, SLOT(setCurrentIndex(1)));
connect(m_okBtn, SIGNAL(clicked()), this, SLOT(activateLicense()));
connect(m_snvnaBtn, SIGNAL(clicked()), this, SLOT(runSNVNA()));

}

```

```

MainWindow::~MainWindow()

```

```

{
    delete ui;
}

```

```

bool MainWindow::checkKey(QString key)

```

```

{
    QSqlQuery query(m_db);

    query.prepare("SELECT * FROM planarcustomers WHERE ключ = :key;");
    query.bindValue(":key", key);
    query.exec();
    int count = 0;
    while(query.next())
    {
        count++;
        m_licenseType = query.value(4).toString();
    }
}

```

```
m_contractNumber = query.value(5).toString();
m_expireDate = query.value(6).toString();
}
if(count == 1)
{
    return true;
}
else
{
    return false;
}
}

void MainWindow::activateLicense()
{
    QString key = m_keyLineEdit->text();
    if(checkKey(key))
    {
        m_licenseLabel->setText("Приветствуем! Ваш тариф: " + m_licenseType + ", срок
окончания: " + m_expireDate);
        m_snvnaBtn->setEnabled(true);
        m_okBtn->setDisabled(true);
        m_keyLineEdit->setDisabled(true);
        m_activateBtn->setDisabled(true);
        m_contactLabel->setText("Ваш номер договора " + m_contractNumber + ".
Сообщите его администратору.");
    }
    else
    {
```

QMessageBox::warning(this, "Ошибка", "Вы ввели неверный ключ. Попробуйте еще раз. Если не получается — обратитесь к нашему администратору по телефону +7(123) 456-78-90 или по почте admin@mail.com. Сообщите администратору название вашей компании или номер договора.");

```

        m_keyLineEdit->clear();
    }
}

void MainWindow::runSNVNA()
{
    qDebug() << "Запуск программы SNVNA";
}

```

Листинг Б2. mainwindow.h.

```

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QStackedLayout>
#include <QLineEdit>
#include <QLabel>
#include <QPushButton>
#include <QMessageBox>
#include <QDateTime>
#include <QSqlDatabase>
#include <QSqlQuery>
#include <QSqlRecord>
#include <QSqlError>
#include <QSqlTableModel>
#include <QTableView>

```

```
QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private:
    Ui::MainWindow *ui;
    QWidget *m_widget;

    QLabel *m_label;
    QStackedLayout *m_layout;

    QWidget *m_productsWidget;
    QLabel *m_productsLabel;

    QWidget *m_activateWidget;
    QPushButton *m_activateBtn;
    QLineEdit *m_keyLineEdit;
    QLabel *m_licenseLabel;
    QPushButton *m_okBtn;

    QLabel *m_infoLabel;
    QPushButton *m_svnBtn;
```

```
QLabel *m_contactLabel;

QSqlDatabase m_db;

QString m_licenseType;
QString m_contractNumber;
QString m_expireDate;

bool checkKey(QString key);

private slots:
    void activateLicense();
    void runSNVNA();
};

#endif // MAINWINDOW_H
```