

Муромский институт (филиал)
федерального государственного бюджетного образовательного учреждения высшего образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(МИ ВлГУ)

Факультет	ИТР
Кафедра	ПИН

КУРСОВАЯ РАБОТА

по курсу Системы управления базами данных
на тему: «АИС учёта программного обеспечения»

К. Т. Н., доц. каф. ПИН
(уч. степень, звание)

Колпаков А.А.
(фамилия, инициалы)

(дата)

Студент _____ ПИН-122
(група)

Корякин А.А.
(фамилия, инициалы)

(подпись)

(дата)

Члены комиссии

(подпись)

(Φ.Π.Ο.)

(подпись)

(Φ.Π.Ο.)

(оценка)

Муром 2024

Данная курсовая работа посвящена разработке автоматизированной информационной системы (АИС) учёта программного обеспечения. В процессе разработки будут рассмотрены основные функциональные возможности АИС, которые позволят эффективно осуществлять учет, контроль и анализ использования программного обеспечения в учебной организации.

Проект включает в себя 37 страниц, 12 литературных источников и 9 приложений.

Работа состоит из введения, четырёх параграфов и заключения.

The present course work is dedicated to the development of an automated information system (AIS) for software accounting. During the development process, the main functional capabilities of the AIS will be examined, which will enable effective accounting, monitoring, and analysis of software usage in an educational organization.

The project contains 37 pages, 12 literary sources and 9 appendices.

The work consists of an introduction, four paragraphs and a conclusion.

Содержание

Введение.....	5
1. Анализ технического задания.....	6
2. Разработка моделей данных.....	8
3. Разработка программного кода приложения	11
3.1. Разработка SQL-запросов.....	13
3.2. Руководство программиста	15
3.3. Руководство пользователя	21
4. Тестирование.....	23
Заключение	27
Список литературы.....	28
Приложения.....	29

					МИВУ 09.03.04					
Изм.	Лист	№ докум.	Подпись	Дата	Программа для учёта ПО			Лит.	Лист	Листов
Разраб.		Корякин А.А.								
Провер.		Колпаков А.А.							4	37
Реценз.								МИ ВлГУ ПИН-122		
Н. Контр.										
Утверд.		.								

Введение

В современных условиях стремительного развития информационных технологий и цифровизации образовательного процесса, эффективное управление программным обеспечением (ПО) становится одной из ключевых задач для учебных организаций. Автоматизированные информационные системы (АИС) играют важную роль в оптимизации процессов учета, контроля и анализа использования ПО, что, в свою очередь, способствует повышению качества образовательных услуг и улучшению организационной структуры.

Учебные заведения, включая школы, колледжи и университеты, сталкиваются с необходимостью учета большого количества программных продуктов, которые используются как преподавателями, так и студентами. Это включает в себя лицензирование ПО, мониторинг его использования, а также обеспечение соответствия требованиям законодательства и стандартам безопасности. В связи с этим, создание и внедрение АИС для учета программного обеспечения становится актуальной задачей, позволяющей автоматизировать процессы и минимизировать риски, связанные с несанкционированным использованием ПО.

Целью данной работы является разработка концепции автоматизированной информационной системы учета программного обеспечения для учебной организации. В рамках исследования будут рассмотрены основные функции и модули системы, а также предложены рекомендации по ее внедрению и эксплуатации. Особое внимание будет уделено анализу существующих решений на рынке, а также выявлению требований пользователей к функционалу системы.

Таким образом, данная курсовая работа направлена на решение актуальной проблемы учета программного обеспечения в учебных организациях и разработку эффективного инструмента для оптимизации данного процесса.

					МИВУ 09.02.03	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

1. Анализ технического задания

Задание – разработать программу для учёта программного обеспечения

1. Программа используется в учебной организации, в распоряжении которой имеется множество компьютеров, предназначенных для учебного процесса и работы сотрудников организации
2. Компьютеры могут входить в состав компьютерных классов или использоваться сотрудниками единолично
3. Необходимо хранить информацию о том, на каком компьютере установлено какое-либо программное обеспечение, характеристики программного обеспечения (вид лицензии, ключ, при необходимости и т.д.), характеристики компьютера (аппаратные характеристики, расположение, номер, IP-адрес)
4. Должны генерироваться отчеты об установленном ПО по аудиториям, по назначению, по категории ПО
5. В БД предусмотреть хранение изображений (минимум в одном поле) в соответствии с тематикой курсовой работы
6. В БД должны быть реализованы хранимые процедуры и/или триггеры
7. В БД должно быть занесено суммарно не менее 50 записей

Для реализации функциональных возможностей необходимо:

- Реализовать работу с базой данных;
- Разработать структуру базы данных в виде диаграммы «Сущность-связь» (ER-диаграмму)
- Разработать программу с использованием визуальных компонентов.

Исходя из функциональных возможностей, разработка базы данных должна соответствовать следующим требованиям:

- Приложение должно предоставлять возможность просмотра данных, возможность добавления, удаления и изменения данных о программном обеспечении, разработчиках, компьютерах.
- Система подачи заявок на установку программного обеспечения, а также их просмотр
- Система определения возможности установки ПО на конкретный компьютер

					МИВУ 09.02.03	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

- Программа должна иметь визуальный интерфейс для работы с базой данных, это достигается путем визуальных компонентов;
- Обеспечивать корректность и надежность функционирования программы.

Решение задания выполняется в парадигме ООП, язык программирования – С#. Для хранения данных выбрана СУБД PostgreSQL. Проект, состоящий из множества окон, представляет из себя приложение Windows Forms. При запуске приложения открывается окно входа и авторизации. При авторизации под обычным пользователем открывается окно, в котором можно подавать заявки на установку нужного ПО, а также запрашивать отчёты о заявках. Окна просмотра таблиц базы данных имеют ограниченный функционал для обычного пользователя. При авторизации под администратором открывается окно, в котором можно рассматривать заявки и использовать функцию их рассмотрения, в результате которой выявляется возможность установки ПО на конкретный компьютер. Помимо этого администратор может редактировать все таблицы базы данных. Просматривать их можно в отдельных окнах, в которых выводятся данные соответствующих таблиц.

					МИВУ 09.02.03	Лист
Изм.	Лист	№ докум.	Подпись	Дата		7

2. Разработка моделей данных

В данной работе при построении модели данной предметной области используется модель «сущность-связь»(см. *Рисунок №9*). Сущность – это объект, данные о котором хранятся в базе данных. Данные о сущности хранятся в отношении. Каждая сущность имеет свой набор атрибутов. Атрибуты представляют собой свойства, характеризующие сущность. Пользователь работает с программой, которая включает в себя удобный интерфейс и базу данных. Программа предоставляет данные об установленном программном обеспечении на конкретные компьютеры, данные о разработчиках, а также о заявках на установку ПО. Данные о программном обеспечении содержат в себе название ПО, версию, тип лицензии, дату начала и окончания данной лицензии, идентификатор разработчика, номер устройства, на которое установлено данное ПО, и логотип. Данные о разработчиках содержат в себе наименование разработчика или компании, тип компании и расположение. Данные об устройствах в учебной организации содержат в себе название, установленную операционную систему, ip-адрес, а также объём оперативной памяти. Данные о заявках содержат в себе идентификаторы программного обеспечения, его разработчика, номер устройства, и ФИО пользователя, который отправил заявку. В базе данных также хранятся данные о пользователях, куда входят фамилия, имя, отчество, роль, идентификатор устройства, за которым он может работать, и пароль в зашифрованном виде.

Логическая модель данных представляет собой более детализированное представление структуры данных по сравнению с концептуальной моделью. Она служит связующим звеном между абстрактной концептуальной моделью и физической реализацией базы данных. Логическая модель фокусируется на том, как данные будут организованы и взаимодействовать друг с другом, без учета конкретных технологий или систем управления базами данных (СУБД). (см. *Рисунок №10*)

Методология SADT включает в себя набор методов, правил и процедур, предназначенных для создания функциональной модели объекта в рамках

					МИВУ 09.02.03	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

определенной предметной области. В результате применения этой методологии формируется модель, состоящая из диаграмм и текстовых фрагментов, которые взаимно ссылаются друг на друга. Диаграммы являются основными элементами модели, где все функции базы данных и интерфейсы представлены в виде блоков и линий. Точка соединения линии с блоком указывает на тип интерфейса. Процесс создания SADT-модели начинается с отображения всей программы в виде простых компонентов — одного блока и линий, демонстрирующих взаимодействия с функциями за пределами приложения. (см. Рисунок №11)

Физическая модель предназначена для явного представления таблиц в базе данных.

Система хранит в себе информацию о программном обеспечении, разработчиках, устройствах, на которых установлено какое-либо ПО. Также хранится информация о заявках на установку ПО и пользователях системы. Основные таблицы базы данных описаны ниже:

Таблица “Developer” - разработчик:

- id – идентификатор разработчика
- name – наименование разработчика/компании
- type_of_company – тип компании
- location – расположение

Таблица “Device” – устройство:

- id – идентификатор устройства
- name – название устройства
- OS – операционная система, установленная на данном устройстве
- ip_address – ip-адрес устройства
- RAM – объём оперативной памяти, доступный на данном устройстве

Таблица “Request” - заявка:

- id – идентификатор заявки
- id_software – идентификатор ПО

					МИВУ 09.02.03	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

- id_developer – идентификатор разработчика
- id_device – идентификатор устройства, на которое необходимо установить данное ПО
- SNM – ФИО пользователя, отправившего заявку

Таблица “Software” – программное обеспечение:

- id – идентификатор ПО
- name – название ПО
- version – версия ПО
- license – тип лицензии ПО
- license_begin – дата начала действия лицензии
- license_end – дата окончания действия лицензии
- id_device – идентификатор устройства, на котором установлено данное ПО
- id_developer – идентификатор разработчика данного ПО
- logo – логотип данного ПО

Таблица “User” – пользователь:

- id – идентификатор пользователя
- surname – фамилия пользователя
- name – имя пользователя
- middlename – отчество пользователя
- login – логин пользователя
- role – роль пользователя в системе
- id_device – идентификатор устройства
- password_hash – пароль в захешированном виде

Физическая модель базы данных «Учёт ПО»(см. Рисунок №12)

					МИВУ 09.02.03	Лист
Изм.	Лист	№ докум.	Подпись	Дата		10

3. Разработка программного кода приложения

Приложение “Учёт ПО” состоит из множества форм:

- Форма входа в учётную запись
- Форма авторизации в системе
- Форма рабочего пространства обычного пользователя
- Форма рабочего пространства администратора базы данных
- Форма для показа и редактирования таблицы ПО
- Форма для показа и редактирования таблицы разработчиков
- Форма для показа и редактирования таблицы устройств
- Форма для показа и редактирования таблицы пользователей

Форма входа в учётную запись содержит 2 текстовых поля для указания системе конкретного пользователя:

- Логин
- Пароль

На форме также располагаются кнопки для входа в систему и перехода на форму авторизации.

Форма авторизации содержит 6 текстовых полей для указания полей создаваемого пользователя

- Фамилия
- Имя
- Отчество
- Роль
- Логин
- Пароль
- ID компьютера

Вторым визуальным компонентом формы является кнопка для авторизации пользователя.

На форме рабочего пространства обычного пользователя располагается текст с указанием инициалов пользователя, 2 текстовых поля с указанием названия ПО и устройства для занесения данной информации в заявку

					МИВУ 09.02.03	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

пользователя. Следующими компонентами формы являются кнопки для подачи заявки и получения отчёта о данной заявке. Для отображения отчёта на форме располагается большое текстовое поле.

Форма рабочего пространства администратора базы данных содержит текст с указанием инициалов пользователя, пользовательский элемент управления “dataGridView” для просмотра таблицы заявок, и кнопки для анализа и удаления выбранной заявки. На форме также располагается меню выбора для перехода к формам просмотра и редактирования таблиц ПО, разработчиков, устройств и пользователей системы.

Формы для просмотра и редактирования таблиц содержат пользовательский элемент управления “dataGridView” для просмотра соответствующих таблиц, кнопки “Добавить”, “Обновить”, “Удалить” для редактирования этих таблиц, а также текстовые поля для ввода информации, касающейся полей таблиц. На форме просмотра и редактирования таблицы ПО также имеется кнопка для выбора файла с изображением логотипа программного обеспечения.

					МИВУ 09.02.03	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

3.1. Разработка SQL-запросов

SQL (Structured Query Language) — это стандартный язык программирования, используемый для управления и манипулирования реляционными базами данных. SQL позволяет выполнять различные операции, такие как создание, чтение, обновление и удаление данных (операции CRUD).

SQL-запросы можно разделить на несколько категорий:

- DML (Data Manipulation Language): Эти запросы используются для манипуляции данными в таблицах.
 - SELECT: используется для извлечения данных из одной или нескольких таблиц.
 - INSERT: добавляет новые записи в таблицу.
 - UPDATE: изменяет существующие записи в таблице.
 - DELETE: удаляет записи из таблицы.
- DDL (Data Definition Language): Эти запросы используются для определения структуры базы данных.
 - CREATE: создает новые таблицы, индексы или базы данных.
 - ALTER: изменяет структуру существующих объектов базы данных.
 - DROP: удаляет таблицы или другие объекты из базы данных.
- DCL (Data Control Language): Эти запросы управляют доступом к данным.
 - GRANT: предоставляет пользователям права доступа к объектам базы данных.
 - REVOKE: отменяет права доступа у пользователей.

В базе данных “Учёт ПО” используются основные SQL-запросы для добавления, обновления, удаления и получения данных из таблиц (DML-операции):

Листинг №1 – inserting.sql

```
INSERT INTO "Developer" (name, type_of_company, location) VALUES (@name, @toc, @location);  
INSERT INTO "Device" (name, "OS", ip_address, "RAM") VALUES (@name, @os, @ip, @ram);
```

					МИВУ 09.02.03	Лист
Изм.	Лист	№ докум.	Подпись	Дата		13

```

INSERT INTO "Request"(id_software, id_developer, id_device, "SNM") VALUES
(@id_soft, @id_dev, @id_device, @snm);
INSERT INTO "User"(surname, name, middlename, role, id_device, password_hash)
VALUES (@surname, @name, @middlename, @role, @id_device, @password_hash);

```

Листинг №2 – deleting.sql

```

DELETE FROM "Developer" WHERE id = @id;
DELETE FROM "Software" WHERE id = @id;
DELETE FROM "User" WHERE id = @id;
DELETE FROM "Device" WHERE id = @id;

```

Листинг №3 – updating.sql

```

UPDATE "Software" SET name = @name, version = @version, license = @license,
license_begin = @lb, license_end = @le, id_device = @id_device, id_developer =
@id_dev, logo = @logo WHERE id = @id;
UPDATE "User" SET surname = @sn, name = @name, middlename = @mn, role = @role,
id_device = @id_device WHERE id = @id;
UPDATE "Device" SET name = @name, "OS" = @os, ip_address = @ip, "RAM" = @ram
WHERE id = @id;
UPDATE "Developer" SET name = @name, type_of_company = @toc, location =
@location WHERE id = @id;

```

Листинг №3 – selecting.sql

```

SELECT * FROM "User";
SELECT * FROM "Software";

```

В базе данных также имеются запросы на создание вторичных ключей для связывания между собой таблиц(см. Приложение №...)

					МИВУ 09.02.03	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

3.2. Руководство программиста

Данный программный продукт предназначен для учёта программного обеспечения в учебной организации. Для выполнения данной цели приложение должно взаимодействовать с базой данных, где хранятся все необходимые объекты.

Условия эксплуатации программы

- Наличие свободной оперативной памяти 20 МБ
- Операционная система Windows 7 и выше
- Процессор (программа не является требовательной к ресурсам процессора, поэтому подойдет любой оптимальный для ОС)
- Свободное место на диске: 149 КБ
- Устройства ввода: клавиатура, мышь
- Устройства вывода: монитор

Основные характеристики программы

Приложение разработано с помощью платформы .NET 8.0, в среде разработки Visual Studio 2022. SoftwareAccounting.exe является исполняемым файлом программы. В проекте содержатся несколько файлов с исходным кодом на языке C#:

- AdminForm.cs – содержит класс AdminForm, предназначенный для описания событий, происходящих на форме AdminForm
- Autorization.cs – содержит класс Autorization, предназначенный для описания событий, происходящих на форме Autorization
- ClientForm.cs – содержит класс ClientForm, предназначенный для описания событий, происходящих на форме ClientForm
- DataBase.cs – содержит класс DataBase, в котором описаны методы для работы с базой данных через SQL-запросы
- DBSettings.cs – содержит статический класс DBSettings, в котором хранится поле со строкой подключения к базе данных
- Developer.cs – содержит класс Developer, описывающий абстрактное представление таблицы Developer из базы данных

					МИВУ 09.02.03	Лист
Изм.	Лист	№ докум.	Подпись	Дата		15

- DeveloperData.cs – содержит класс DeveloperData, предназначенный для описания событий, происходящих на форме DeveloperData
- Device.cs – содержит класс Device, описывающий абстрактное представление таблицы Device из базы данных
- DeviceData.cs - содержит класс DeviceData, предназначенный для описания событий, происходящих на форме DeviceData
- Login.cs - содержит класс Login, предназначенный для описания событий, происходящих на форме Login
- Request.cs - содержит класс Request, описывающий абстрактное представление таблицы Request из базы данных
- Software.cs - содержит класс Software, описывающий абстрактное представление таблицы Software из базы данных
- SoftwareData.cs - содержит класс SoftwareData, предназначенный для описания событий, происходящих на форме SoftwareData
- SqlCommand.cs – содержит перечисление SqlCommand, описывающее команды SQL
- User.cs - содержит класс User, описывающий абстрактное представление таблицы User из базы данных
- UserData.cs - содержит класс UserData, предназначенный для описания событий, происходящих на форме UserData

Основным классом, обеспечивающим связь между приложением и базой данных является DataBase.

Назначение, структура входных и выходных данных программных функций класса DataBase:

```
public class DataBase
```

Поля

```
private NpgsqlConnection Connection;
```

Хранит экземпляр объекта NpgsqlConnection, обеспечивающего соединение с базой данных

```
private readonly string ConnectionString;
```

Хранит в себе строку подключения с базой данных

					МИВУ 09.02.03	Лист
Изм.	Лист	№ докум.	Подпись	Дата		16

Методы

```
public int Connect()
```

Параметры

-

Назначение

Открывает подключение поля Connection

Возвращаемое значение

```
int
```

Возвращает 0, если открытие подключения успешно выполнилось, иначе -1

```
public void CreateUser(User user)
```

Параматеры

```
user User
```

Экземпляр класса User, хранящий в себе поля таблицы User

Назначение

Заносит нового пользователя в таблицу User

Возвращаемое значение

```
void
```

```
public bool IsExists(User user)
```

Параматеры

```
user User
```

Экземпляр класса User, хранящий в себе поля таблицы User

Назначение

Проверяет, существует ли данный пользователь в таблице User

Возвращаемое значение

```
bool
```

Возвращает true, если user существует в таблице, иначе false

```
public bool IsExists(Software software, bool isFull)
```

Параматеры

```
software Software
```

Экземпляр класса Software, хранящий в себе поля таблицы Software

```
isFull bool
```

Флаг, обозначающий, как именно проверять наличие software в базе данных

Назначение

					МИВУ 09.02.03	Лист
						17
Изм.	Лист	№ докум.	Подпись	Дата		

Проверяет, существует ли данное ПО в таблице Software

Возвращаемое значение

bool

Возвращает true, если software существует в таблице, иначе false

```
public bool IsExists(Developer developer)
```

Параметры

developer Developer

Экземпляр класса Developer, хранящий в себе поля таблицы Developer

Назначение

Проверяет, существует ли данный разработчик в таблице Developer

Возвращаемое значение

bool

Возвращает true, если developer существует в таблице, иначе false

```
public bool IsExists(Device device)
```

Параметры

device Device

Экземпляр класса Device, хранящий в себе поля таблицы Device

Назначение

Проверяет, существует ли данное устройство в таблице Device

Возвращаемое значение

bool

Возвращает true, если device существует в таблице, иначе false

```
public DataSet GetTable(string tableName)
```

Параметры

tableName string

Имя таблицы, записи которой нужно получить

Назначение

Получает все записи указанной таблицы

Возвращаемое значение

DataSet

Возвращает DataSet, хранящий в себе набор данных указанной таблицы

```
public void EditRequest(Request request, SqlCommand sqlCommand)
```

Параметры

request Request

					МИВУ 09.02.03	Лист
						18
Изм.	Лист	№ докум.	Подпись	Дата		

Экземпляр класса Request, хранящий в себе поля таблицы Request

sqlCmd SqlCommand

Объект типа enum, указывающий на то, какую SQL-команду нужно выполнить

Назначение

Редактирует таблицу Request в зависимости от значения команды

Возвращаемое значение

void

```
public void EditDeveloper(Developer dev, SqlCommand sqlCmd)
```

Параметры

dev Developer

Экземпляр класса Developer, хранящий в себе поля таблицы Developer

sqlCmd SqlCommand

Объект типа enum, указывающий на то, какую SQL-команду нужно выполнить

Назначение

Редактирует таблицу Developer в зависимости от значения команды

Возвращаемое значение

Void

```
public void EditSoftware(Software soft, SqlCommand sqlCmd)
```

Параметры

soft Software

Экземпляр класса Software, хранящий в себе поля таблицы Software

sqlCmd SqlCommand

Объект типа enum, указывающий на то, какую SQL-команду нужно выполнить

Назначение

Редактирует таблицу Software в зависимости от значения команды

Возвращаемое значение

Void

```
public void EditDevice(Device device, SqlCommand sqlCmd)
```

Параметры

device Device

Экземпляр класса Device, хранящий в себе поля таблицы Device

sqlCmd SqlCommand

Объект типа enum, указывающий на то, какую SQL-команду нужно выполнить

Назначение

					МИВУ 09.02.03	Лист
						19
Изм.	Лист	№ докум.	Подпись	Дата		

Редактирует таблицу Device в зависимости от значения команды

Возвращаемое значение

Void

```
public void EditUser(User user, SqlCommand sqlCommand)
```

Параметры

user User

Экземпляр класса User, хранящий в себе поля таблицы User

sqlCmd SqlCommand

Объект типа enum, указывающий на то, какую SQL-команду нужно выполнить

Назначение

Редактирует таблицу User в зависимости от значения команды

Возвращаемое значение

void

```
public List<string> GetLastRequest(Request request)
```

Параметры

request Request

Экземпляр класса Request, хранящий в себе поля таблицы Request

Назначение

Получает данные последней заявки и возвращает их в виде списка

Возвращаемое значение

List<string>

Список строк, в которых хранятся имена ПО, устройства и разработчика

```
public void Disconnect()
```

Параметры

-

Назначение

Закрывает подключение к базе данных

Возвращаемое значение

void

					МИВУ 09.02.03	Лист
						20
Изм.	Лист	№ докум.	Подпись	Дата		

3.3. Руководство пользователя

Данная система представляет из себя прикладную программу с множеством окон, а также базу данных, в которой хранится вся необходимая информация о пользователях системы и программном обеспечении, установленном на компьютеры учебной организации.

При запуске программы пользователь видит окно входа/авторизации в систему. Для входа в учётную запись нужно ввести логин и пароль, а затем нажать на кнопку “Войти”. Чтобы зарегистрироваться в системе, нужно перейти на форму регистрации по нажатию на кнопку “Регистрация”. На форме нужно ввести ФИО, логин, пароль, выбрать устройство, на котором работает пользователь, и отметить роль, а затем нажать на кнопку “Зарегистрироваться”.

В зависимости от роли пользователя система открывает форму рабочего пространства обычного пользователя в случае роли “user”, или форму рабочего пространства администратора, если роль пользователя “admin”. Форма рабочего пространства обычного пользователя позволяет подавать заявки на установку ПО, а также получать отчёты о поданной заявке. Чтобы подать заявку, нужно выбрать доступное программное обеспечение из первого списка выбора, а также выбрать устройство, на которое нужно установить данное ПО. Если все компоненты выбраны, нужно нажать на кнопку “Подать заявку”, после чего данные заявки занесутся в базу данных. После этого можно получить отчёт о заявке, нажав на кнопку “Получить отчёт”. Отчёт отобразится на нижнем текстовом поле в разделе “Отчёты”.

Форма рабочего пространства администратора позволяет просматривать текущие заявки пользователей, которые отображаются в таблице заявок на форме. Выбрав определённую заявку, можно провести анализ, в ходе которого система определит, можно ли установить ПО на компьютер, указанные в выбранной заявке. Система также определяет, установлено ли уже данное ПО на указанное устройство. Чтобы провести анализ заявки, нужно нажать на кнопку “Анализ заявки”. Помимо данной функции администратору предоставляется удаление нужной заявки по нажатию на кнопку “Удалить заявку”. Ещё одной функцией

					МИВУ 09.02.03	Лист
Изм.	Лист	№ докум.	Подпись	Дата		21

является отметка тех заявок, которые уже рассмотрены. Для этого в таблице заявок в колонке “Состояние” имеются флажки, на которые можно нажать, тем самым отметив заявку рассмотренной.

Форма рабочего пространства администратора, кроме анализа заявок, позволяет перейти на формы просмотра и редактирования таблиц базы данных. Перейти к ним можно посредством меню выбора “База данных” в верхней части формы. На данных формах администратор может просматривать таблицы ПО, разработчиков, устройств и пользователей системы, а также редактировать их. Для этого на формах располагаются текстовые поля с нужными полями. Чтобы отредактировать таблицу, нужно выбрать конкретную запись, нажав на нее, после чего её данные занесутся в нужные поля, и выбрать нужное действие путём нажатия на кнопки “Добавить”, “Обновить”, “Удалить”.

Иллюстрация приложения представлена на рисунках №13-16

					МИВУ 09.02.03	Лист
Изм.	Лист	№ докум.	Подпись	Дата		22

4. Тестирование

Важным завершающим этапом разработки программного продукта является процесс тестирования и отладки. На этом этапе осуществляется комплексная проверка функциональности программы, что позволяет выявить ее недостатки и оценить особенности работы. Тестирование играет ключевую роль в обеспечении качества и надежности приложения, позволяя разработчикам устранить возможные ошибки и повысить общую производительность системы.

На данном этапе будут тестироваться основные функции приложения, такие как добавление, обновление, удаление записей из таблиц, функция анализа заявок, проверка существования пользователей и т.п.

Тест №1 – Проверка существования пользователя при входе в систему.

Попробуем войти в систему под несуществующим пользователем

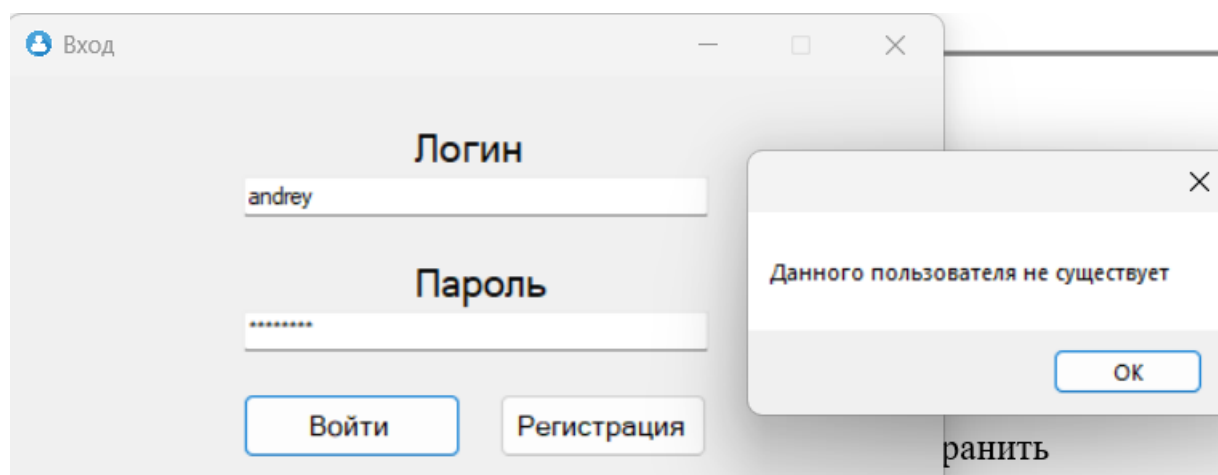


Рисунок №1 – Попытка входа в систему под несуществующим пользователем

Система выдала сообщение о том, что пользователя с логином “andrey” не существует. В данном случае такого пользователя действительно нет в системе, поэтому тест можно считать успешным

Тест №2 – Анализ заявок

Попробуем провести анализ заявки, в которой указано ПО, уже установленное на устройство, которое так же указано в данной заявке

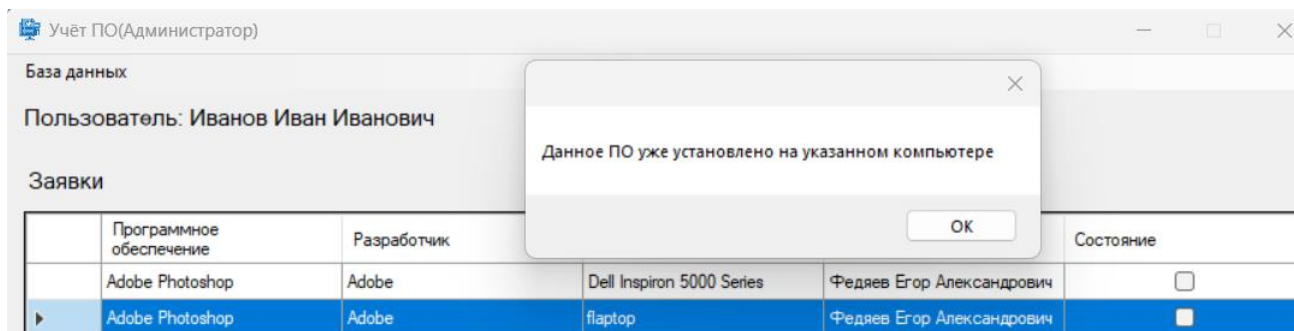


Рисунок №2 – Анализ заявки

Программное обеспечение								
	Название	Версия	Лицензия	Начало лицензии	Завершение лицензии	Разработчик	Устройство	Логотип
▶	Adobe Photoshop	v1.0	Коммерческая	01.01.2024	01.01.2028	Adobe	flaptop	
	Matlab	v1.3	Коммерческая	08.09.2023	08.09.2030	MathWorks	Dell Inspiron 5000 Series	
	JDK	v23.5	Свободная	01.01.2021	01.01.2025	Oracle	ASUS VivoBook	
	JDK	v23.5	Свободная	01.01.2021	01.01.2025	Oracle	Dell Inspiron 5000 Series	

Рисунок №3 – Таблица установленных ПО

Программное обеспечение “Adobe Photoshop”, установленное на устройство “flaptop”, имеется в таблице Software, поэтому функция анализа заявки выдала соответствующее сообщение

Тест №3 – Добавление записи в таблицу

Добавим в таблицу устройств компьютер, ip-адрес которого уже имеется в таблице

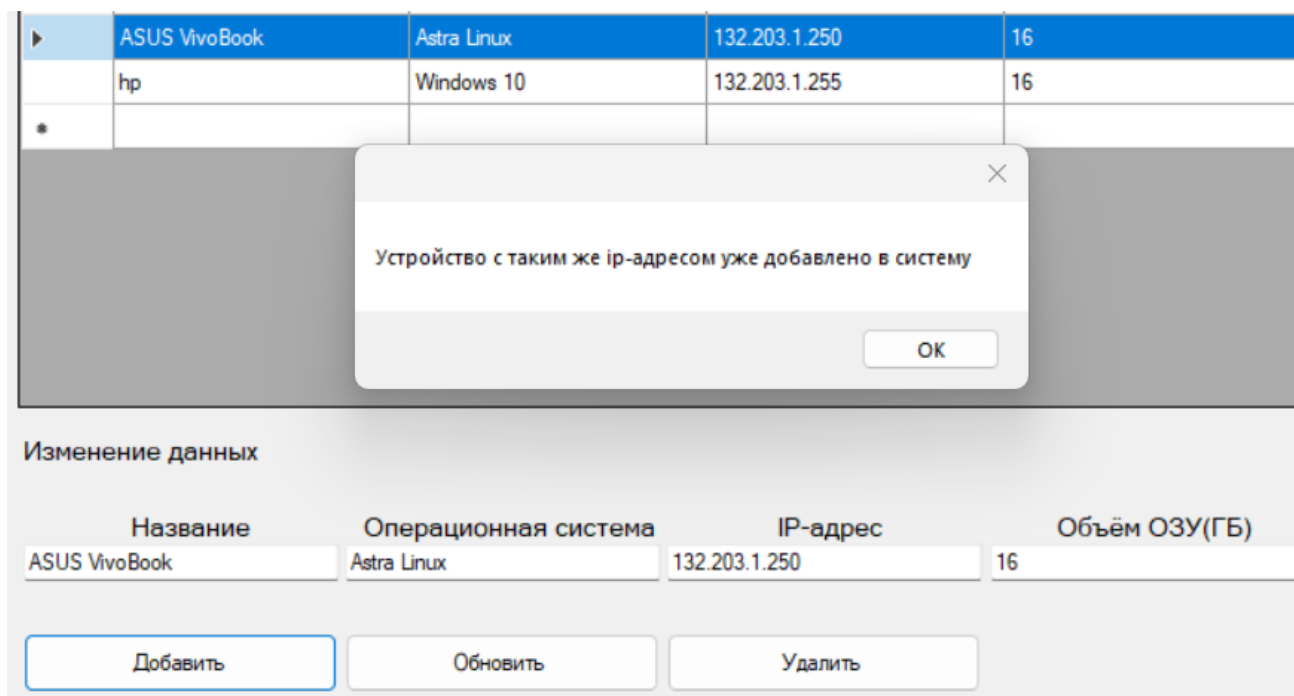


Рисунок №4 – Добавление устройства с существующим ip-адресом

На рисунке №4 видно, что система обнаружила устройство с таким же ip-адресом, что и в добавляемом устройстве, поэтому сообщила об ошибке

Тест №4 – Обновление записи в таблице

В таблице разработчиков обновим конкретную запись

Разработчики			
	Название	Тип компании	Расположение
	Microsoft	Corporation	Redmond, Washington, USA
▶	Oracle	Corporation	Redwood City, California, USA

Рисунок №5 – Выбор записи

	Oracle	Private company	Redwood City, California, USA
*			

Изменение данных

Название	Тип компании	Расположение
Oracle	Private company	Redwood City, California, USA

Добавить

Обновить

Удалить

Рисунок №6 – Обновлённая запись

На рисунках №5-6 видно, что запись удачно обновилась

Тест №5 – Удаление записи из таблицы

В таблице пользователей удалим одну запись

Пользователи						
	Фамилия	Имя	Фамилия	Логин	Роль	Доступ к устройству
	Иванов	Иван	Иванович	ivan	admin	2, Dell Inspiron 5000 Series
	Андреев	Андрей	Викторович	andrew	user	1, laptop
	Федяев	Егор	Александрович	iam	user	3, HP Pavilion Series
▶	Александров	Данил	Дмитриевич	dan12	user	1, laptop

Рисунок №7 – Выбор пользователя

Пользователи

	Фамилия	Имя	Фамилия	Логин	Роль	Доступ к устройству
▶	Иванов	Иван	Иванович	ivan	admin	2, Dell Inspiron 5000 Series
	Андреев	Андрей	Викторович	andrew	user	1, laptop
	Федяев	Егор	Александрович	iam	user	3, HP Pavilion Series
*						

Изменение данных

Фамилия: Александров
 Имя: Данил
 Отчество: Дмитриевич
 Роль: user
 Доступ к устройству: 1, laptop

Рисунок №8 – Удаление выбранного пользователя

На рисунках №7-8 видно, что пользователя с логином “dan12” после удаления в таблице нет

Заключение

В результате данной работы была создана АИС “Учёт программного обеспечения”. Данная система использует базу данных, в которой хранится информация об установленном ПО в некоторой учебной организации. Программа позволяет пользователям авторизоваться, составлять и отправлять заявки на установку ПО, получать отчёты. Администраторы могут проводить анализ заявок, а также редактировать таблицы.

В процессе работы была создана архитектура базы данных, состоящая из разных моделей данных, а также разработано настольное приложение, предоставляющее определённый функционал.

Подводя итоги, можно считать, что разработанная АИС соответствует требованиям технического задания.

					МИВУ 09.02.03	Лист
Изм.	Лист	№ докум.	Подпись	Дата		27

Список литературы

1. Бочаров, В. И. Информационные системы и технологии: учебник для вузов / В. И. Бочаров, А. В. Шеремет. — М.: Юрайт, 2020. — 480 с.
2. Глущенко, О. В. Системы автоматизированного учёта: теория и практика / О. В. Глущенко. — СПб.: БХВ-Петербург, 2021. — 352 с.
3. Костромина, Е. В. Программное обеспечение и его учёт в корпоративных информационных системах / Е. В. Костромина. — М.: Дело, 2019. — 250 с.
4. Петров, Н. Н. Информационные технологии в учёте и контроле программного обеспечения / Н. Н. Петров. — М.: Академия, 2021. — 300 с.
5. Смирнов, А. А. Программное обеспечение для учёта и мониторинга: анализ и разработка / А. А. Смирнов. — М.: Наука, 2020. — 180 с.
6. Николаев, В. П. Автоматизация учёта программного обеспечения на предприятиях / В. П. Николаев. — М.: Инфра-М, 2018. — 150 с.
7. ГОСТ Р 56026-2014. Программное обеспечение. Управление жизненным циклом программных продуктов. — М.: Стандартинформ, 2014.
8. Барсов, И. В. Управление информационными системами: теоретические основы и практика / И. В. Барсов. — М.: ЮНИТИ-ДАНА, 2017. — 450 с.
9. Соловьев, В. И. Основы информационных систем и технологий: учебник / В. И. Соловьев. — М.: Высшая школа, 2020. — 432 с.
10. Иванов, С. Л. Современные подходы к учёту и лицензированию программного обеспечения / С. Л. Иванов. — СПб.: Бизнес-Пресс, 2019. — 220 с.
11. Леонтьев, Д. В. Разработка и внедрение систем учёта программного обеспечения на предприятии / Д. В. Леонтьев. — СПб.: Питер, 2021. — 160 с.
12. Системы управления базами данных: учебник для вузов / Под ред. А. В. Никифорова. — М.: Издательство МГУ, 2018. — 480 с.

					МИВУ 09.02.03	Лист
Изм.	Лист	№ докум.	Подпись	Дата		28

Ссылка на репозиторий проекта: <https://github.com/thendontuser/Software-Accounting>

					МИВУ 09.02.03	Лист
						29
Изм.	Лист	№ докум.	Подпись	Дата		

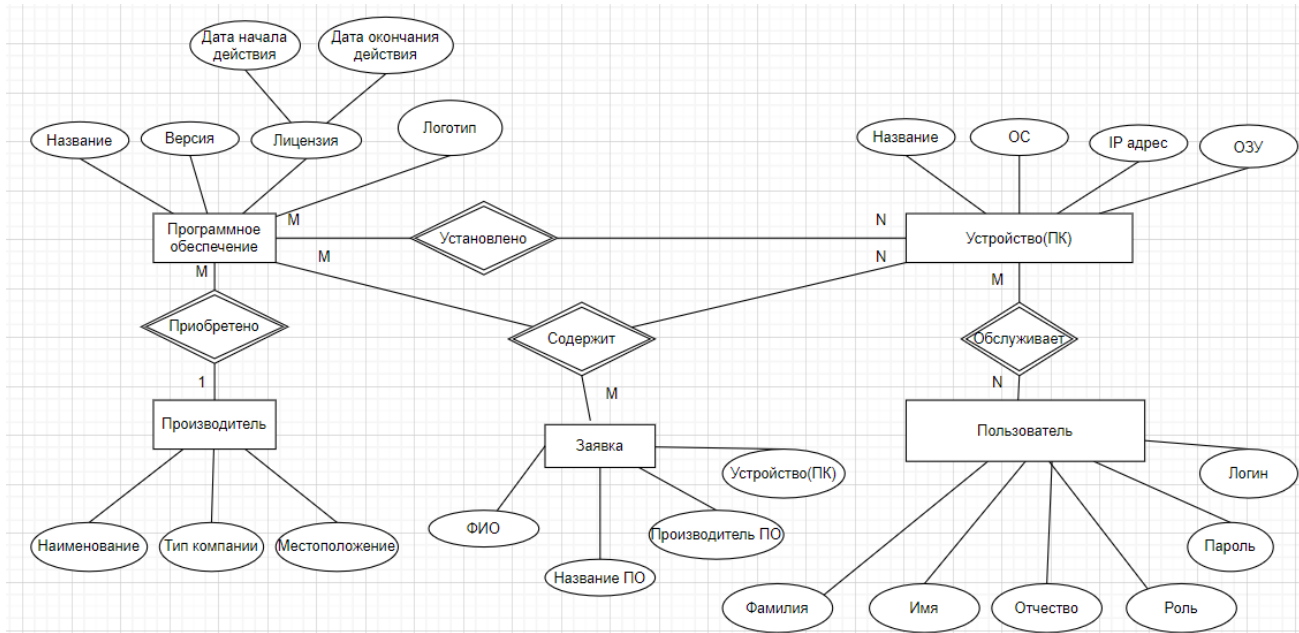


Рисунок №9 – Диаграмма “сущность-связь”

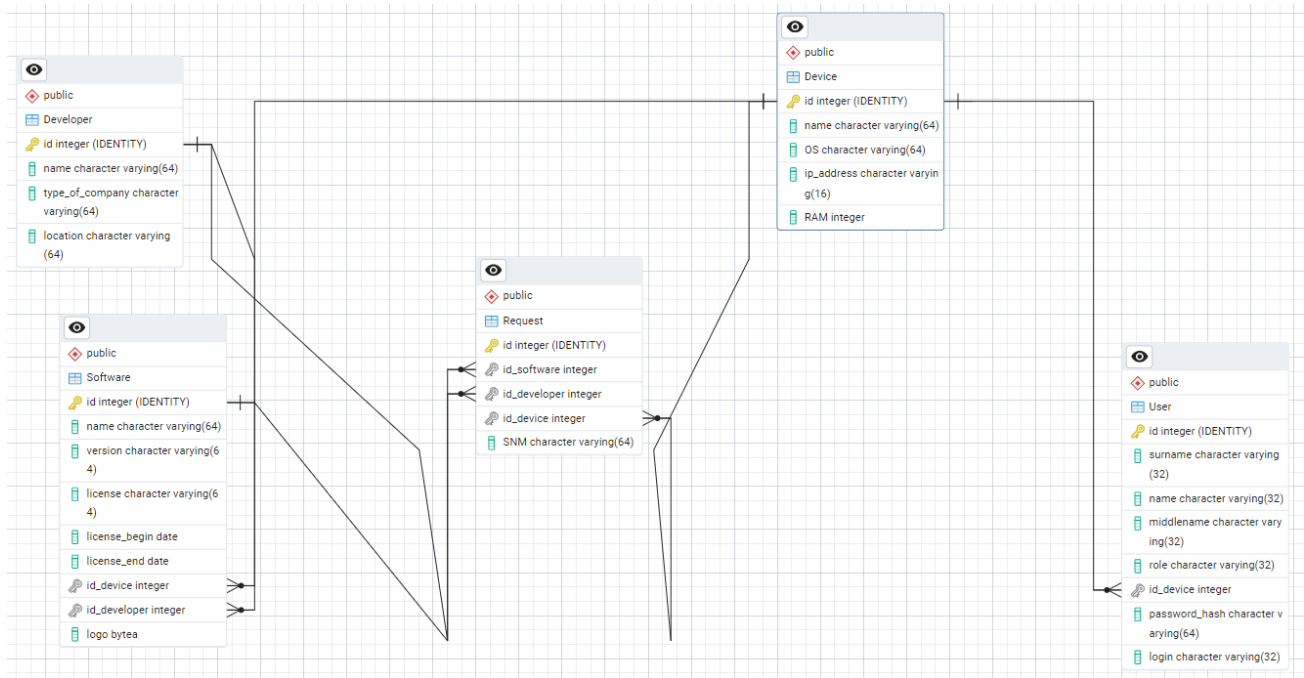


Рисунок №10 – Логическая модель данных

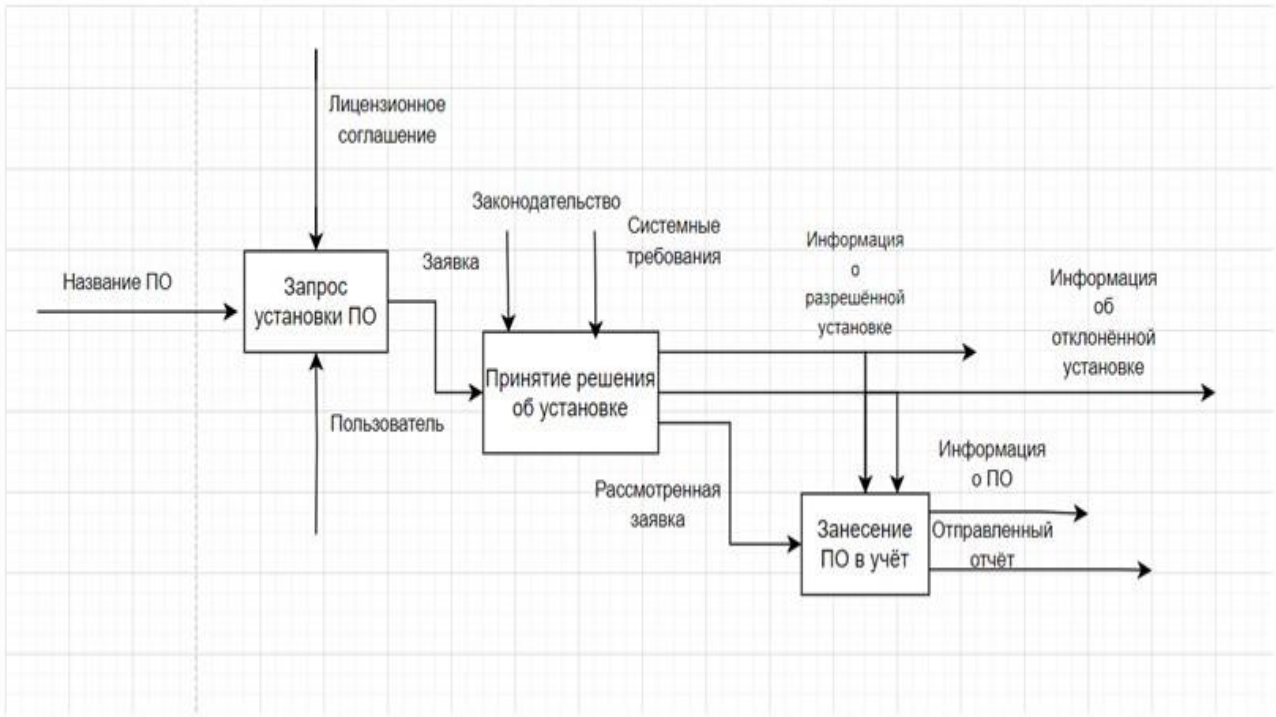


Рисунок №11 – SADT-диаграмма

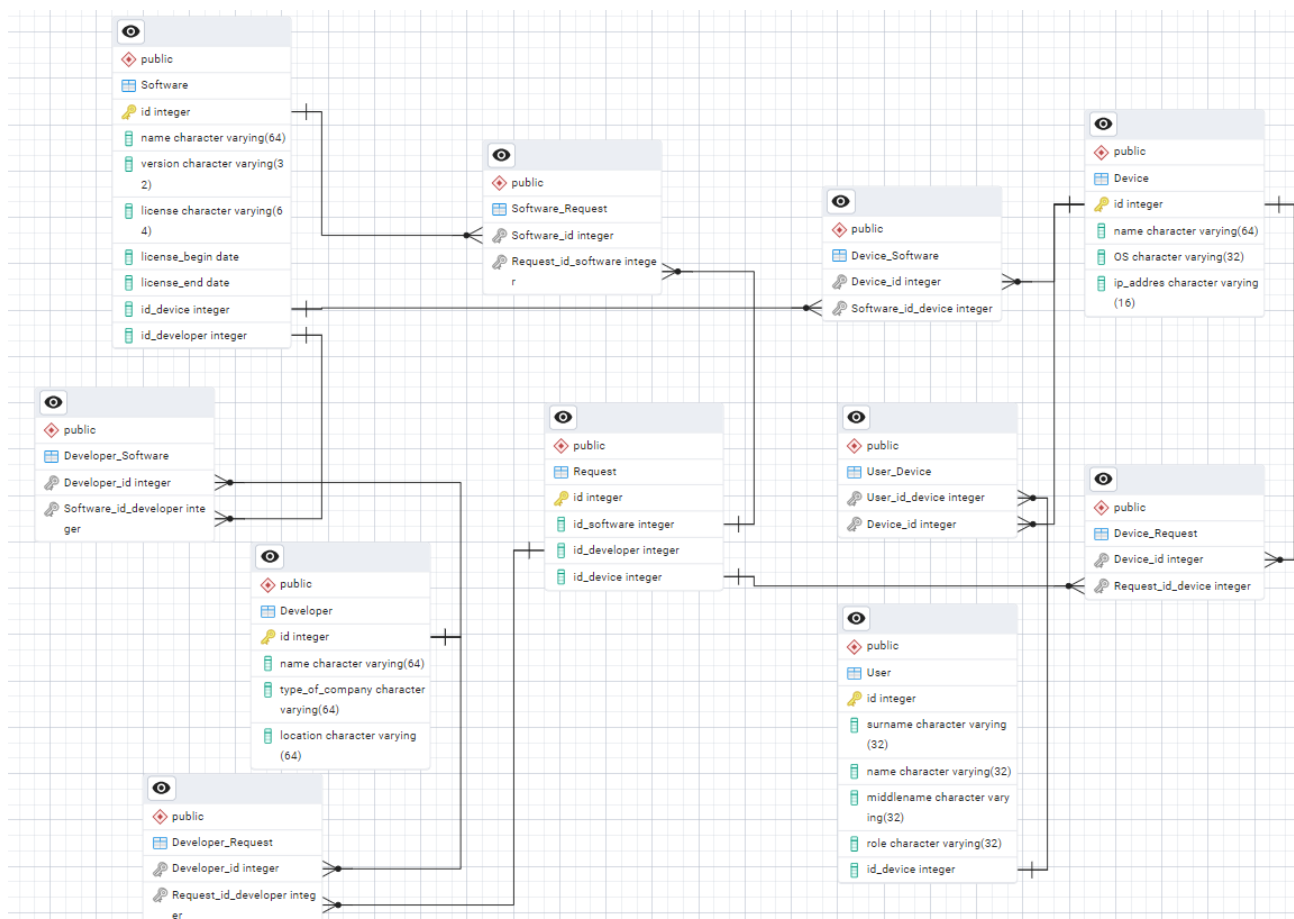


Рисунок №12 – Физическая модель данных

Вход

Логин

Пароль

Войти Регистрация

Рисунок №13 – Окно авторизации приложения

Учёт ПО(Клиент)

Пользователь: Федяев Егор Александрович

Данные заявки

Укажите название ПО:

Укажите название устройства:

Отчёты

Рисунок №14 – Рабочее пространство обычного пользователя

Учёт ПО(Администратор)

База данных

Пользователь: Иванов Иван Иванович

Заявки









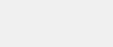

	Программное обеспечение	Разработчик	Устройство	ФИО	Состояние
▶	Adobe Photoshop	Adobe	Dell Inspiron 5000 Series	Федяев Егор Александрович	<input type="checkbox"/>
	Adobe Photoshop	Adobe	flaptop	Федяев Егор Александрович	<input type="checkbox"/>
	JDK	Oracle	Asus XP	Федяев Егор Александрович	<input type="checkbox"/>
	Matlab	MathWorks	Asus 2 Series	Федяев Егор Александрович	<input type="checkbox"/>
	Adobe Photoshop	Adobe	Asus XP	Федяев Егор Александрович	<input type="checkbox"/>
	Adobe Photoshop	Adobe	flaptop	Федяев Егор Александрович	<input type="checkbox"/>
	Matlab	MathWorks	Dell Inspiron 5000 Series	Федяев Егор Александрович	<input type="checkbox"/>
	JDK	Oracle	hp	Федяев Егор Александрович	<input type="checkbox"/>
	Visual Studio	Microsoft	Asus 2 Series	Федяев Егор Александрович	<input type="checkbox"/>
*					<input type="checkbox"/>

Анализ заявки

Удалить заявку

Рисунок №15 – Рабочее пространство администратора

Программное обеспечение

	Название	Версия	Лицензия	Начало лицензии	Завершение лицензии	Разработчик	Устройство	Логотип
▶	Adobe Photoshop	v1.0	Коммерческая	01.01.2024	01.01.2028	Adobe	flaptop	
	Matlab	v1.3	Коммерческая	08.09.2023	08.09.2030	MathWorks	Dell Inspiron 5000 Series	
	JDK	v23.5	Свободная	01.01.2021	01.01.2025	Oracle	ASUS VivoBook	
	JDK	v23.5	Свободная	01.01.2021	01.01.2025	Oracle	Dell Inspiron 5000 Series	
	Adobe Photoshop	v1.0	Коммерческая	01.01.2024	01.01.2028	Adobe	hp	
	Sublime Text	3	Свободная	01.01.2025	01.01.2050	SublimeDev	HP Pavilion Series	
	Visual Studio	2022	Свободная	01.01.2022	01.01.2027	Microsoft	Dell Inspiron 5000 Series	
	Visual Studio	2022	Свободная	01.01.2022	01.01.2027	Microsoft	HP Pavilion Series	
	JDK	v23.5	Свободная	01.01.2021	01.01.2025	Oracle	hp	
*								

Изменение данных

Название:
 Версия:
 Лицензия:
 Дата начала действия лицензии:
 Дата окончания действия лицензии:
 Разработчик:
 Установлено на устройстве:

Логотип:

Рисунок №16 – Форма просмотра и редактирования таблицы ПО