**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ**

**ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ**

**(АКТ (ф) СПбГУТ)**

СОГЛАСОВАНО

Рук. предприятия

\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

(Подпись) (И.О. Фамилия)

«31» мая 2025г.

**ТЕХНИЧЕСКИЙ ОТЧЁТ**

**по ПМ.11, ПМ.01**

|  |
| --- |
| УФК ПО АРХАНГЕЛЬСКОЙ ОБЛАСТИ И НАО |
| Информационные системы и программирование |
| 09.02.07. 25ТО01. 024 ПЗ |
| (Обозначение документа) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИСПП-21 | |  | 31.05.2025 | Н.Д. Попов |
|  | (Группа) | | (Подпись) | (Дата) | (И.О. Фамилия) |
| Рук. практики от предприятия | | |  | 31.05.2025 | А.В. Денисов |
|  | |  | (Подпись) | (Дата) | (И.О. Фамилия) |

Архангельск 2025

Содержание

[Перечень сокращений и обозначений 3](#_Toc200835789)

[Введение 4](#_Toc200835790)

[1 Охрана труда и техника безопасности при работе на ПК 6](#_Toc200835791)

[1.1 Общие требования безопасности 6](#_Toc200835792)

[1.2 Требования безопасности во время работы 6](#_Toc200835793)

[2 Выполнение работ по ПМ.11 8](#_Toc200835794)

[2.1 Проектирование базы данных 8](#_Toc200835795)

[2.2 Разработка базы данных и объектов базы данных 10](#_Toc200835796)

[2.3 Администрирование и защита базы данных 13](#_Toc200835797)

[3 Выполнение работ по ПМ.01 16](#_Toc200835798)

[3.1 Проектирование ПО 16](#_Toc200835799)

[3.2 Разработка программных модулей 17](#_Toc200835800)

[3.3 Разработка мобильного приложения 18](#_Toc200835801)

[3.4 Отладка и тестирование программных модулей 20](#_Toc200835802)

[3.5 Оптимизация и рефакторинг программного кода 21](#_Toc200835803)

[Заключение 23](#_Toc200835804)

[Список использованных источников 24](#_Toc200835805)

Перечень сокращений и обозначений

В настоящем техническом отчете применяются следующие сокращения и обозначения:

БД – база данных

ВДТ – видео-дисплейный терминал

ИС – информационная система

ОС – операционная система

ПК – персональный компьютер

ПО – программное обеспечение

ПЭВМ – персональная электронно-вычислительная машина

СУБД – система управления базами данных

УФК – управление федерального казначейства

НАО – непубличное акционерное общество

API **–** программный интерфейс приложения

DDL – язык описания данных

ORM - объектно-реляционное отображение

RAM – оперативная память

SQL – язык структурированных запросов

UI – графический интерфейс

UML – унифицированный язык моделирования

WPF – Windows Presentation Foundation

Введение

Базой производственной практики является Управление Федерального казначейства по Архангельской области и Ненецкому автономному округу, осуществляющее приём, распределение, учёт и контроль бюджетных средств с применением современных информационных технологий.

Цели производственной практики является:

* получение практического опыта по выполнению работ по ПМ.01 «Разработка программных модулей программного обеспечения для компьютерных систем» и развитие общих и профессиональных компетенций;
* получение практического опыта по выполнению работ по ПМ.11 «Разработка, администрирование и защита баз данных» и развитие общих и профессиональных компетенций.

Задачи производственной практики являются:

* формирование алгоритмов разработки ПМ в соответствии с техническим заданием;
* разработка ПМ в соответствии с техническим заданием, с использованием современных языков программирования и сред разработки;
* выполнение отладки ПМ с использованием специализированных программных средств;
* выполнение тестирования ПМ;
* осуществление рефакторинга и оптимизации программного кода;
* разработка модулей ПО для мобильных платформ;
* осуществление сбора, обработки и анализа информации для проектирования БД;
* проектирование баз данных в соответствии с результатами анализа предметной области, с учётом требований к структуре, целостности и производительности;
* разработка объектов БД в соответствии с результатами анализа предметной области;
* реализация БД в конкретной СУБД;
* администрирование БД;
* обеспечение защиты информации в базах данных с использованием современных технологий и методов информационной безопасности.

Для практикантов предоставляется рабочее место с персональным компьютером и всем необходимым для работы аппаратным и программным обеспечением:

* процессор: Intel Core i3-9100 3.6GHz;
* системная плата: MSI H310M Pro-VDH Plus;
* видеокарта: встроенная;
* оперативная память – 8ГБ;
* операционная система: Microsoft Windows 10 Home;
* прикладное ПО: пакет Microsoft Office, Яндекс Браузер, Visual Studio 2022, Microsoft SQL Server Management Studio 18, draw.io.

Кроме рабочего места также предоставляется доступ к серверу предприятия со всем необходимым для работы аппаратным и программным обеспечением:

* процессор: Intel Pentium CPU G3240 3.1GHz;
* системная плата: MSI H83M-P33;
* видеокарта: встроенная;
* оперативная память – 4ГБ;
* операционная система: Microsoft Windows Server 2016 Essentials;
* прикладное ПО: Microsoft SQL Server Express 2019.

# Охрана труда и техника безопасности при работе на ПК

## Общие требования безопасности

Инструкция применяется ко всем работникам, использующим ПЭВМ и ВДТ. К самостоятельной работе допускаются лица старше 18 лет, прошедшие медицинский осмотр и обучение, а также инструктаж по электробезопасности. Для беременных женщин время работы с ПЭВМ не должно превышать трех часов в день.

## Требования безопасности во время работы

Пользователь обязан поддерживать чистоту рабочего места, не закрывать вентиляцию и корректно завершать задачи. Запрещается прикасаться к задней панели системного блока при включенном питании, закрывать оборудование бумагами и отключать питание во время активной задачи. Рабочие режимы должны соответствовать гигиеническим требованиям, с обязательными перерывами для физических упражнений.

Перед началом работы необходимо убедиться в исправности оборудования: монитор, клавиатура, системный блок, мышь и кабели не должны иметь внешних повреждений. Подключение оборудования к электросети должно производиться только через исправные розетки, предпочтительно с заземлением. Запрещается использовать удлинители с признаками износа, а также размещать провода в местах, где возможна их деформация или перегиб.

Во время работы следует избегать чрезмерного наклона корпуса над монитором, соблюдать рекомендуемое расстояние до экрана (от 50 до 70 см) и располагать его таким образом, чтобы избежать бликов от внешнего освещения. Место должно быть хорошо освещено, предпочтительно естественным светом, но без попадания прямых солнечных лучей на экран. Важно соблюдать правильную осанку: стопы должны стоять на полу, спина – опираться на спинку стула, а запястья не должны находиться в постоянном напряжении.

Пользователь должен строго следить за температурным режимом и проветриванием помещения. Перегрев оборудования может привести к его выходу из строя, а также создаёт угрозу возгорания. При перегреве или появлении характерного запаха гари работа должна быть немедленно прекращена, оборудование отключено от питания, и об этом следует сообщить ответственному специалисту или системному администратору.

Запрещается открывать корпус системного блока или другого оборудования без соответствующего разрешения. Все технические работы должны проводиться только квалифицированными специалистами. Также недопустимо самостоятельно устранять неисправности, особенно связанные с электрическими соединениями, блоком питания и жёсткими дисками.

Для защиты информации и предотвращения несанкционированного доступа важно использовать пароли и регулярно их менять. Не допускается использование внешних накопителей (флешек, дисков) без предварительной проверки на вирусы. Следует избегать посещения сомнительных сайтов и загрузки программного обеспечения из ненадёжных источников.

# Выполнение работ по ПМ.11

## Проектирование базы данных

Управление Федерального казначейства по Архангельской области и Ненецкому автономному округу является государственным учреждением, в структуре которого активно используется компьютерная техника и программное обеспечение. Подразделения и сотрудники ежедневно работают с различными IT-ресурсами, что требует централизованного учёта оборудования, пользователей, установленных программ и возникающих инцидентов.

Цель базы данных – обеспечить контроль за IT-активами, упростить поддержку и анализировать инциденты. Система особенно полезна для IT-отделов в средних и крупных компаниях для управления обращениями пользователей и лицензиями ПО.

На рисунке 1 показана концептуальная модель предметной области в виде ERD, созданная с помощью средства проектирования Draw.io.

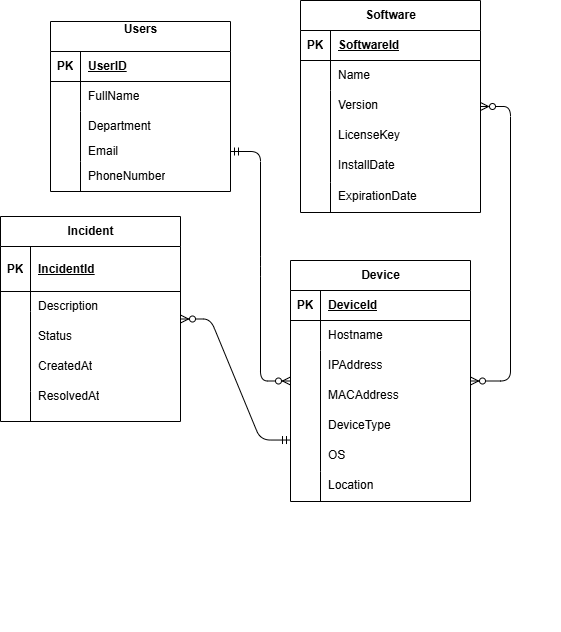


Рисунок 1 – Концептуальная модель

СУБД Microsoft SQL Server, выбранная для хранения данных, является реляционной, поэтому разработанная концептуальная модель преобразуется в логическую с учётом правил перехода от ER-модели к реляционной модели данных.

При преобразовании учитываются типы связей между сущностями:

* связь между Device и Users – M:1 и необязательная для Users, поэтому формируется два отношения с внешним ключом в отношении Device;
* связь между Software и Device – M:M, поэтому формируется три отношения с внешними ключами с промежуточным отношением SoftwareDevice;
* связь между Incident и Device – M:1 и необязательная для Device, поэтому формируется одно отношение с внешним ключом в отношении Incident.

На рисунке 2 показана логическая модель предметной области, полученная в результате преобразования концептуальной модели, которая Fсоздана с использованием средства визуального проектирования Draw.io. Логическая модель спроектирована по принципу третьей нормальной формы и обеспечивает минимизацию избыточности данных. [3]

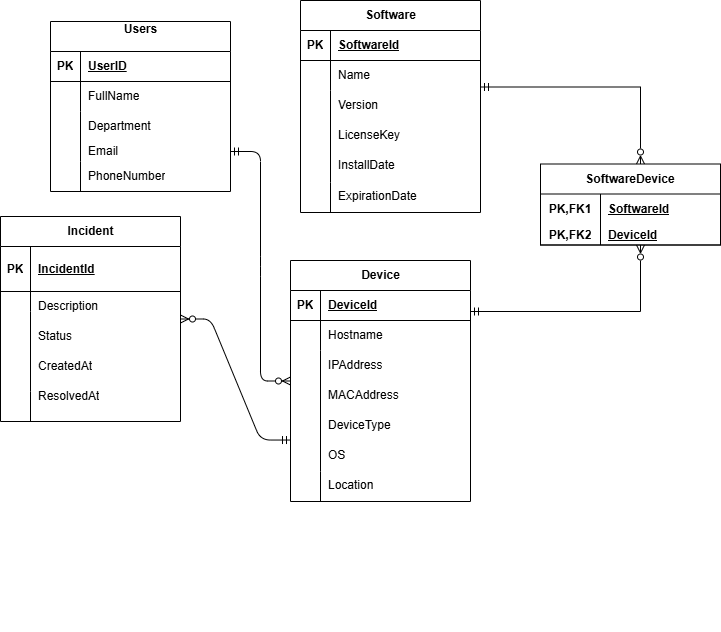


Рисунок 2 – Логическая модель

На рисунке 3 представлена физическая модель предметной области, разработанная на основе ранее сформированной логической модели. Данная модель отражает конкретную реализацию проектируемой базы данных в среде СУБД Microsoft SQL Server Express 2021.

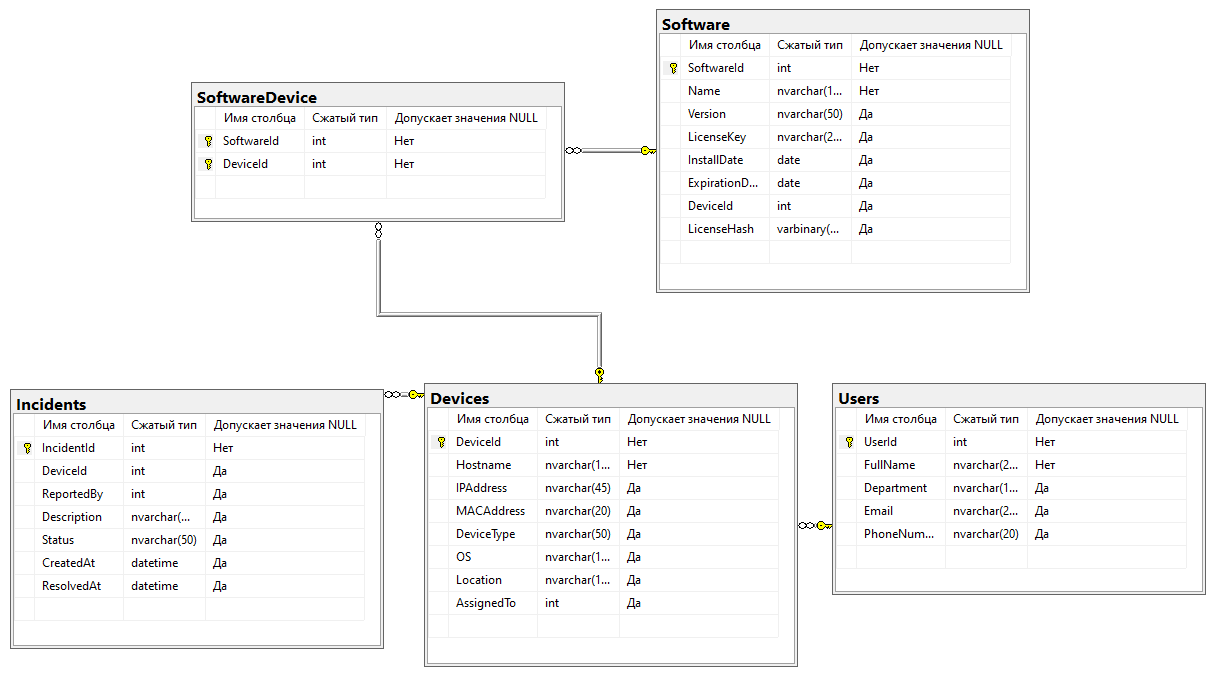


Рисунок 3 – Физическая модель

## Разработка базы данных и объектов базы данных

В таблице 1 в виде словаря данных представлено описание созданных таблиц и ограничений целостности БД.

Таблица 1 – Словарь данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ключ** | **Поле** | **Тип данных** | **Обязательное** | **Примечание** |
| Incidents | | | | |
| PK | IncidentId | int | + | Автоинкрементный |

*Продолжение таблицы 1*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ключ** | **Поле** | **Тип данных** | **Обязательное** | **Примечание** |
| FK1 | DeviceId | int | + | Внешний ключ – Devices(DevicesId) |
| FK2 | ReportedBy | int | + | Внешний ключ – Users(UserId) |
|  | Description | nvarchar(MAX) | + |  |
|  | Status | nvarchar(50) | + | По умолчанию ‘Open’ |
|  | ResolvedAt | datetime | - | Null, если не решено |
|  | CreatedAt | datetime | + | По умолчанию GETDATE() |

Для создания таблиц БД и ограничений целостности требуется выполнить соответствующие DDL команды. Код создания таблицы Incidents представлен листингом 1.

Листинг 1 – SQL-запрос для создания таблицы Incidents

--Создание таблицы Incidents с полями и первичным ключом

CREATE TABLE [dbo].[Incidents](

[IncidentId] [int] IDENTITY(1,1) NOT NULL,

[DeviceId] [int] NULL,

[ReportedBy] [int] NULL,

[Description] [nvarchar](max) NULL,

[Status] [nvarchar](50) NULL,

[CreatedAt] [datetime] NULL,

[ResolvedAt] [datetime] NULL,

PRIMARY KEY CLUSTERED

(

[IncidentId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

Для решения поставленных задач для таблиц БД требуется создать уникальны ограничения в соответствии со словарем данных, используя команды DDL. Код создания уникальных ограничений для таблиц представлен листингом 2.

Листинг 2 – Код создания уникальных ограничений

--Создание уникального ограничения для таблицы Software

GO

ALTER TABLE Software

ADD CONSTRAINT [UQ\_Software] UNIQUE ([Name])

GO

Для решения поставленных задач в БД требуется создать скалярную функцию GetDeviceIncidentCount, которая возвращает количество инцидентов, зарегистрированных по конкретному устройству. Код создания функции представлен в листинге 3.

Листинг 3 – Код создания скалярной функции GetDeviceIncidentCount

--Функция возвращает количество инцидентов, зарегистрированных по конкретному устройству.

CREATE FUNCTION GetDeviceIncidentCount

(

@DeviceId INT

)

RETURNS INT

AS

BEGIN

DECLARE @IncidentCount INT

SELECT @IncidentCount = COUNT(\*)

FROM Incidents

WHERE DeviceId = @DeviceId

RETURN @IncidentCount

END

Для удобного отображения сведений об установленном программном обеспечении, включая устройства и пользователей, требуется создать представление SoftwareWithDeviceAndUser. Оно объединяет данные из таблиц Software, Devices и Users, позволяя видеть полную цепочку установки ПО. Код создания представления представлен в листинге 4.

Листинг 4 – Представление SoftwareWithDeviceAndUser

-- Представление отображает сведения о ПО, устройстве и закреплённом пользователе

CREATE VIEW SoftwareWithDeviceAndUser AS  
SELECT  
s.Name AS SoftwareName,  
s.Version,  
s.LicenseKey,  
d.Hostname,  
d.IPAddress,  
u.FullName AS AssignedUser  
FROM Software s  
JOIN Devices d ON s.DeviceId = d.DeviceId  
JOIN Users u ON d.AssignedTo = u.UserId;

Для обеспечения контроля за регистрацией инцидентов требуется реализовать триггер, предотвращающий добавление записей с пустым описанием. Это позволяет избежать ситуаций, когда в таблицу Incidents вносятся неконкретные или ошибочные обращения. Код создания триггера представлен в листинге 5.

Листинг 5 – Триггер trg\_BlockEmptyIncidentDescription

-- Блокирует вставку инцидентов с пустым описанием

CREATE TRIGGER trg\_BlockEmptyIncidentDescription  
ON Incidents  
AFTER INSERT  
AS  
BEGIN  
SET NOCOUNT ON;

IF EXISTS (

SELECT 1 FROM inserted WHERE LTRIM(RTRIM(ISNULL(Description, ''))) = ''

)

BEGIN

RAISERROR('Описание инцидента не может быть пустым.', 16, 1);

ROLLBACK TRANSACTION;

RETURN;

END

Одним из этапов настройки системы является заполнение ключевых таблиц начальными данными, выполнен ручной ввод информации об устройстве с помощью базовой операции INSERT INTO. Код добавления записи в таблицу Devices представлен в листинге 6.

Листинг 6 – Код для добавления устройства в таблицу Devices

INSERT INTO Devices ([DeviceId]

,[Hostname]

,[IPAddress]

,[MACAddress]

,[DeviceType]

,[OS]

,[Location]

,[AssignedTo])

VALUES ('5', 'DSK-MNG01', '192.168.1.50', '00-11-22-33-44-99', 'Desktop','Windows 11 Pro', 'Офис директора', '5');

## Администрирование и защита базы данных

В целях повышения информационной безопасности в системе реализовано хеширование лицензионных ключей с использованием алгоритма SHA2-256. Это обеспечивает защиту данных от несанкционированного доступа и подделки.

SHA2-256 – современный криптографический алгоритм, формирующий 256-битное хеш-значение, которое невозможно восстановить в исходный вид. Лицензионные ключи из таблицы Software преобразуются и сохраняются в поле LicenseHash с помощью SQL-функции HASHBYTES. При этом:

* значение приводится к типу NVARCHAR;
* хеш создаётся через SHA2-256;
* оригинальный ключ не сохраняется и может быть проверен только путём повторного хеширования.

Такая архитектура хранения повышает защищённость базы данных от утечки информации и соответствует рекомендациям по безопасной разработке. Пример SQL-запроса, выполняющего хеширование всех лицензионных ключей, приведён в листинге 7.

Листинг 7 – Код реализации алгоритма хеширования на языке SQL

SET LicenseHash = HASHBYTES('SHA2\_256', CAST(LicenseKey AS NVARCHAR(255)));

INSERT INTO Software (Name, Version, LicenseKey, LicenseHash)

Для ограничения доступа к данным требуется создать отдельные логины и пользователи базы данных. Сотрудникам назначаются права только на чтение, тогда как технические специалисты получают расширенный доступ (чтение и изменение).

Для создания логинов, пользователей и назначения прав необходимо выполнить набор SQL-команд, представленный в листинге 8.

Листинг 8 – Создание логина и пользователя для сотрудника с правом только чтения

-- Листинг 6: создание логина и пользователя для сотрудника с правом только чтения

CREATE LOGIN User\_ReadOnly WITH PASSWORD = 'UserPassword123!';

CREATE USER User\_ReadOnly FOR LOGIN User\_ReadOnly;

EXEC sp\_addrolemember 'db\_datareader', 'User\_ReadOnly';

Во избежание потери данных необходимо периодически выполнять полное резервное копирование БД. Резервные копии позволяют восстановить данные после сбоя и других непредвиденных проблем. Для выполнения резервного копирования БД требуется выполнить SQL-скрипт, представленный листингом 9.

Листинг 9 – Код для создания резервной копии БД IT\_Inventory

--Выполнение резервного копирования

BACKUP DATABASE [IT\_Inventory] TO

DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\IT\_Inventory.bak'

WITH NOFORMAT, NOINIT,

NAME = N'IT\_Inventory-Резервное копирование',

SKIP, NOREWIND, NOUNLOAD, STATS = 10;

Для обеспечения надёжного хранения и восстановления данных в случае сбоев или потери информации предусмотрено создание резервных копий базы данных. Одним из ключевых этапов в стратегии защиты данных является возможность восстановления базы из резервной копии.

Данный процесс позволяет восстановить рабочее состояние информационной системы без потери критически важных данных. Восстановление выполняется с использованием специального SQL-скрипта, представленного в листинге 10.

Листинг 10 – Код для восстановления из резервной копии БД

--Восстановление из резервной копии

RESTORE DATABASE IT\_Inventory FROM

DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\IT\_Inventory.bak'

WITH FILE = 2, NOUNLOAD, STATS = 5;

# Выполнение работ по ПМ.01

## Проектирование ПО

Предприятием поставлена задача разработки специализированного программного обеспечения, обеспечивающего следующую функциональность:

* форматирование диска;
* анализ диска;
* логирование действий;
* мониторинг диска.

Разрабатываемое программное обеспечение предназначено для использования специалистами IT-отдела казначейства в целях обслуживания и контроля технического состояния рабочих станций, выявления сбоев, очистки накопителей и фиксации служебных операций. Система ориентирована на внутреннее использование в рамках эксплуатации компьютерной инфраструктуры учреждения.

Действия, доступные пользователю приложения, отображены на диаграмме прецедентов, представленной на рисунке 4.

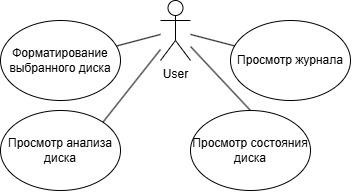


Рисунок 4 – Диаграмма прецедентов

## Разработка программных модулей

В качестве языка разработки использовался C#[1] с использованием технологии WPF для создания пользовательского интерфейса. Для реализации функционала форматирования и выбора диска, разработан класс FormatHelper. Код этого класса приведён в листинге 11.

Листинг 11 – Код класса FormatHelper

public static class FormatHelper

{

public static void FormatDrive(string driveLetter, string fileSystem = "NTFS", bool quickFormat = true)

{ // Формируем команду форматирования

string formatCommand = $"/C format {driveLetter} /FS:{fileSystem} {(quickFormat ? "/Q" : "")} /Y";

ProcessStartInfo psi = new ProcessStartInfo("cmd.exe", formatCommand)

{

Verb = "runas", // Запуск от имени администратора

UseShellExecute = true,

CreateNoWindow = false

};

Process.Start(psi);

}

}

Используя средства WPF [4], требуется спроектировать интерфейс для выбора действий пользователя в соответствии с рисунком 5.

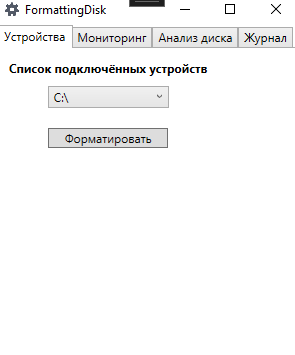


Рисунок 5 – FormattingDisk. Вид основного окна

## Разработка мобильного приложения

Предприятием поставлена задача разработать приложение для удобного создания хранения и быстрого доступа к личным заметкам пользователя. Приложение должно предоставлять следующие возможности:

* создание и редактирование текстовых заметок;
* сохранение данных.

Приложение ориентировано на платформу Android, реализовано на языке Kotlin в среде Android Studio, с использованием современных технологий разработки мобильных интерфейсов.[5]

Главная страница приложения NotesScreen (рисунок 6) cодержит поля для ввода заголовка и содержания, кнопки добавления и удаления записей.

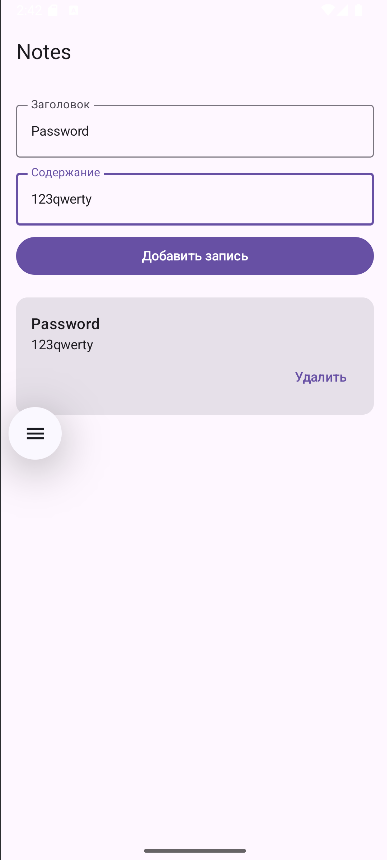


Рисунок 6 – NotesScreen. Вид главной страницы

Пользователь может заполнить заметку, указав заголовок и содержание, а затем нажать кнопку «Добавить запись», расположенную внутри блока Scaffold. В этом блоке размещены элементы интерфейса, включая поля ввода и кнопку, отвечающую за сохранение данных. Логика заполнения формы и обработки кнопки представлена в листинге 12.

Листинг 12 – Логика заполнения и обработки кнопки

Scaffold(  
 topBar = { TopAppBar(title = { Text("Notes") }) },  
 content = { padding ->  
 Column(modifier = Modifier  
 .padding(padding)  
 .padding(16.dp)) {  
  
 OutlinedTextField(  
 value = title,  
 onValueChange = { title = it },  
 label = { Text("Заголовок") },  
 modifier = Modifier.fillMaxWidth()  
 )  
 Spacer(modifier = Modifier.height(8.dp))  
 OutlinedTextField(  
 value = content,  
 onValueChange = { content = it },  
 label = { Text("Содержание") },  
 modifier = Modifier.fillMaxWidth()  
 )  
 Spacer(modifier = Modifier.height(8.dp))  
  
 Button(  
 onClick = {  
 if (editingId != null) {  
 showUpdateDialog = true  
 } else {  
 scope.launch {  
 viewModel.saveNote(null, title, content)  
 title = ""  
 content = ""  
 editingId = null  
 }  
 }  
 },  
 modifier = Modifier.fillMaxWidth()  
 )

## Отладка и тестирование программных модулей

Для обеспечения стабильной и надёжной работы приложения было проведено тестирование ключевых компонентов с целью выявления логических, функциональных и пользовательских ошибок.

Краткое описание действий, ожидаемых и фактических результатов представлено в таблице 2.

Таблица 2 – Набор тестовых приложений

|  |  |  |
| --- | --- | --- |
| **Действие** | **Ожидаемый результат** | **Полученный результат** |
| Выбрать диск C:\ и нажать кнопку «Форматировать» | Отформатируется диск, вывод результата пользователю об успехе. | Соответствует ожидаемому |
| Выбрать в главном окне «Мониторинг» | Вывод результата состояния дисков. | Соответствует ожидаемому |
| Выбрать в главном окне «Анализ диска», выбрать диск и нажать кнопку «Анализировать диск» | Вывод результата анализа выбранного диска. | Соответствует ожидаемому |
| Выбрать в главном окне «Журнал» и нажать кнопку «Открыть журнал» | Вывод операций программы, который делал пользователь. | Соответствует ожи13даемому |

Также необходимо провести автоматизированное тестирование приложения. Для этого были разработаны unit-тесты, позволяющие проверить корректность работы ключевых компонентов системы. Код тестов приведён в листингах 13 и 14, где представлен код методов тестирования класса FormatHelper, а также проверки логики взаимодействия с файловой системой и логированием действий.

Листинг 13 – Код unit-теста Log\_WritesAndReadsLogSuccessfully()

[Fact]

public void Log\_WritesAndReadsLogSuccessfully()

{

// Arrange

string testMessage = $"Тест логирования {Guid.NewGuid()}";

// Act

Logger.Log(testMessage);

string content = Logger.ReadLog();

// Assert

Assert.Contains(testMessage, content);

}

Листинг 14 – Код unit-теста FormatDrive\_DoesNotThrowException()

[Fact]

public void FormatDrive\_DoesNotThrowException()

{

// Arrange

string testDrive = "Z:"; // Важно: не должен существовать, чтобы не форматировался настоящий диск

// Act & Assert

var ex = Record.Exception(() => FormatHelper.FormatDrive(testDrive, "NTFS", true));

// Проверяем, что не выброшено исключение

Assert.Null(ex);

}

## Оптимизация и рефакторинг программного кода

В процессе разработки приложения «FormattingDisk» была выполнена оптимизация и рефакторинг кода с целью повышения читаемости, производительности и соответствия стандартам программирования.

Проведённые улучшения включали:

* устранение дублирующихся фрагментов кода и вынесение повторяющихся операций в отдельные методы;
* оптимизацию работы с ресурсами;
* повышение отзывчивости пользовательского интерфейса за счёт внедрения асинхронности;
* улучшение читаемости и сопровождения за счёт стандартизации имён.

Для соответствия стилю C# [2] проведена нормализация именования: методы и классы приведены к формату PascalCase, переменные – к camelCase. Пример переименования представлен в листинге 15.

Листинг 15 – Оптимизация вызова анализа диска

// До рефакторинга:  
var x = new DiskInfo();

// После рефакторинга:

var result = new DiskInfo();

Также оптимизирована операция анализа диска, переведённая в асинхронный режим выполнения (листинг 16).

Листинг 15 – оптимизирована операция анализа диска

// До:

var result = DiskAnalyzerMain.AnalyzeDisk(driveLetter);

// После:

var result = await Task.Run(() => DiskAnalyzerMain.AnalyzeDisk(driveLetter));

Заключение

Для прохождения практики в Управлении Федерального казначейства по Архангельской области и Ненецкому автономному округу учреждением было предоставлено всё необходимое оборудование и программное обеспечение.

Цели производственной практики достигнуты:

* получен практический опыт по выполнению работ по ПМ.01 «Разработка программных модулей программного обеспечения для компьютерных систем» и развиты общие и профессиональные компетенции;
* получен практический опыт по выполнению работ по ПМ.11 «Разработка и администрирование баз данных» и развиты общие и профессиональные компетенции.

Для достижения целей практики выполнены следующие задачи:

* формирование алгоритмов разработки ПМ в соответствии с техническим заданием;
* разработка ПМ в соответствии с техническим заданием, с использованием современных языков программирования и сред разработки;
* выполнение тестирования ПМ;
* осуществление рефакторинга и оптимизации программного кода;
* разработка модулей ПО для мобильных платформ;
* осуществление сбора, обработки и анализа информации для проектирования БД;
* проектирование баз данных в соответствии с результатами анализа предметной области, с учётом требований к структуре, целостности и производительности;
* реализация БД в конкретной СУБД;
* администрирование БД;
* обеспечение защиты информации в базах данных с использованием современных технологий и методов информационной безопасности.

Список использованных источников

1. Пахомов Б. И. *C# для начинающих*. – СПб : БХВ‑Петербург, 2014. – 432 с. – URL: <https://lesmatveev.narod.ru/knigi_proect/ci_sharp.pdf> (дата обращения: 31.05.2025). – Свободный доступ.
2. Александрова И. Л., Тумаков Д. Н. *Программирование на языке C#*. – Казань : Казанский федеральный университет, 2017. – 112 с. – URL: <https://kpfu.ru/portal/docs/F304170973/Aleksandrova_I_L___Tumakov_D_N_P.pdf> (дата обращения: 13.05.2025). – Свободный доступ.
3. Рихтер Д**.** CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. – 4-е изд. / Д. Рихтер. – Санкт-Петербург : Питер, 2012. – 896 с. URL: <https://archive.org/download/windowsprogram/CLR_via_CSharp_%28Jeffrey_Richter_4th_Edition%29.pdf> (дата обращения: 20.05.2025). – Режим доступа: свободный. – Текст: электронный.
4. WPF Documentation : [сайт] / корпорация Microsoft. – Редмонд, 2000–2025. – Обновляется ежедневно. – URL: https://learn.microsoft.com/en-us/dotnet/desktop/wpf/?view=netdesktop-8.0 (дата обращения: 18.05.2025). – Текст : электронный.
5. JetBrains. *Документация по Kotlin для Android-разработчиков*. – URL: <https://kotlinlang.org/docs/android-overview.html> (дата обращения: 15.05.2025). – Свободный доступ.