**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**

|  |
| --- |
| ООО “ВЕГА-С” |
| Информационные системы и программирование |
| 09.02.07. 25ТО01. 024 ПЗ |
| (Обозначение документа) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИСПП-21 | |  | 31.05.2025 | В.А. Казнин |
|  | (Группа) | | (Подпись) | (Дата) | (И.О. Фамилия) |
| Рук. практики от предприятия | | |  | 31.05.2025 | А.С. Казнин |
|  | |  | (Подпись) | (Дата) | (И.О. Фамилия) |

Архангельск 2025

Содержание

[Перечень сокращений и обозначений 3](#_Toc201521566)

[Введение 4](#_Toc201521567)

[1 Охрана труда и техника безопасности при работе на ПК 6](#_Toc201521568)

[1.1 Общие требования безопасности 6](#_Toc201521569)

[1.2 Требования безопасности во время работы 6](#_Toc201521570)

[2 Выполнение работ по ПМ.11 8](#_Toc201521571)

[2.1 Проектирование базы данных 8](#_Toc201521572)

[2.2 Разработка базы данных и объектов базы данных 10](#_Toc201521573)

[2.3 Администрирование и защита базы данных 14](#_Toc201521574)

[3 Выполнение работ по ПМ.01 17](#_Toc201521575)

[3.1 Проектирование ПО 17](#_Toc201521576)

[3.2 Разработка программных модулей 18](#_Toc201521577)

[3.3 Разработка мобильного приложения 19](#_Toc201521578)

[3.4 Отладка и тестирование программных модулей 21](#_Toc201521579)

[3.5 Оптимизация и рефакторинг программного кода 22](#_Toc201521580)

[Заключение 24](#_Toc201521581)

[Список использованных источников 25](#_Toc201521582)

Перечень сокращений и обозначений

В настоящем техническом отчете применяются следующие сокращения и обозначения:

БД – база данных

ВДТ – видео-дисплейный терминал

ПК – персональный компьютер

ПМ – программный модуль

ПО – программное обеспечение

СУБД – система управления базами данных

DDL – язык описания данных

SQL – язык структурированных запросов

WPF – Windows Presentation Foundation

Введение

Базой производственной практики является **строительная компания ООО «ВЕГА-С»,** специализирующаяся на проектировании, строительстве и эксплуатации объектов гражданского и промышленного назначения.

Целью 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.

Задачами 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 i3-8100 3.6GHz;
* системная плата: Gigabyte H310M-S2H;
* видеокарта: встроенная;
* оперативная память – 8ГБ;
* операционная система: Microsoft Windows 10 Home;
* прикладное ПО: пакет Microsoft Office, Google Chrome, 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 G4400 3.3GHz;
* системная плата: ASRock H110M-HDS;
* 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

**ООО «ВЕГА-С»** – строительная организация, осуществляющая деятельность в сфере проектирования, снабжения и возведения объектов различного назначения. В процессе производственной деятельности важную роль играет эффективное управление складом строительных материалов, так как своевременная доставка и учёт ресурсов напрямую влияют на соблюдение сроков строительства.

Цель базы данных – организация централизованного учёта остатков, поставок и перемещений товаров на складе. Система позволяет отслеживать приход и расход материалов, автоматизировать инвентаризацию, а также анализировать остатки в разрезе категорий, поставщиков и объектов строительства. Такое решение особенно полезно для небольших и средних строительных компаний.

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

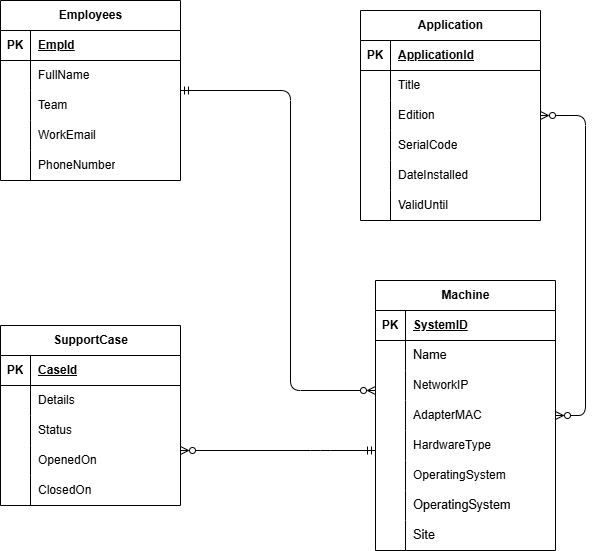


Рисунок 1 – Концептуальная модель

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 Machine и Employees – M:1 и необязательная для Employees, поэтому формируется два отношения с внешним ключом в отношении Machine;
* связь между Application и Machine – M:M, поэтому формируется три отношения с внешними ключами в промежуточном отношении AppSystemMap;
* связь между SupportCase и Machine – M:1 и необязательная для Machine, поэтому формируется два отношения с внешним ключом в отношении Incident.

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

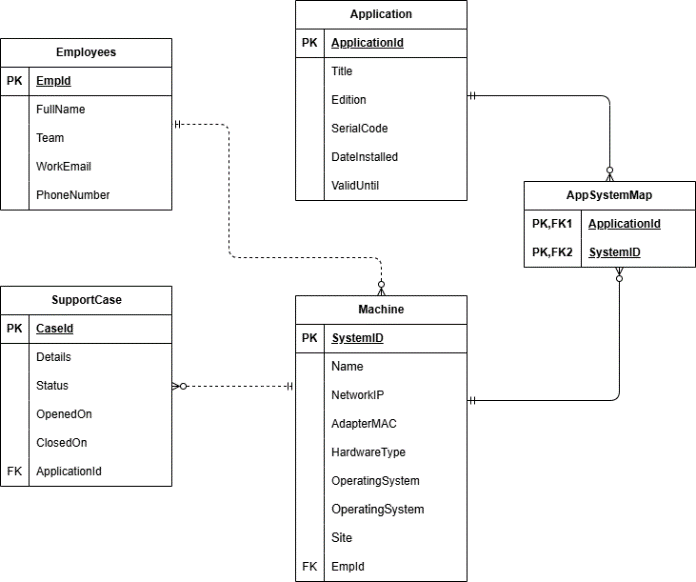


Рисунок 2 – Логическая модель

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.

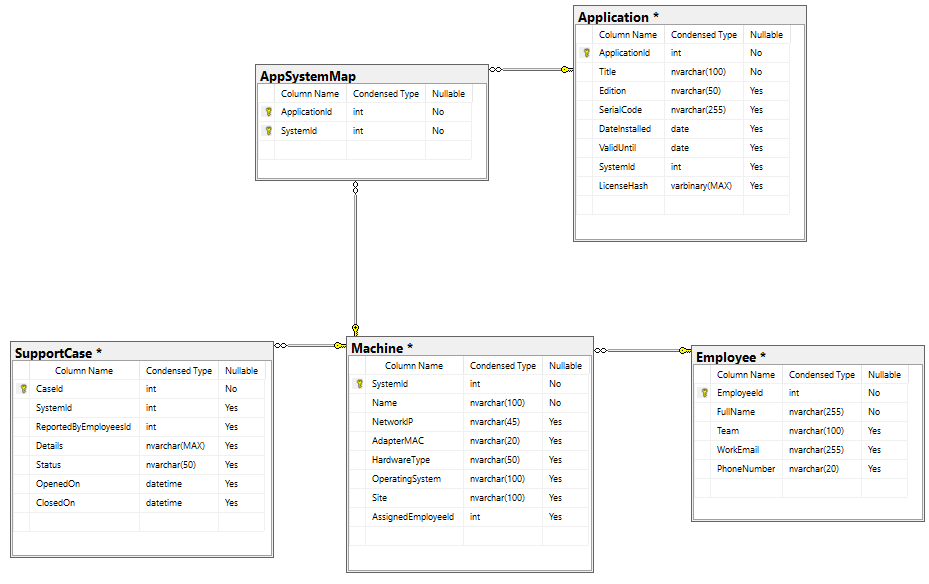


Рисунок 3 – Физическая модель

## 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** |
| Machine | | | | |
| PK | SystemId | int | + | Автоинкрементный |

*Продолжение таблицы 1*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ключ** | **Поле** | **Тип данных** | **Обязательное** | **Примечание** |
|  | Name | nvarchar(100) | + |  |
|  | NetworkIP | nvarchar(45) | + |  |
|  | AdapterMAC | nvarchar(20) | + |  |
|  | HardwareType | nvarchar(50) | + | По умолчанию ‘Open’ |
|  | OperatingSystem | nvarchar(100) | - | Null, если не решено |
|  | Site | nvarchar(100) | + | По умолчанию GETDATE() |
| FK1 | AssignedEmployeeId | int | + | Внешний ключ – Employee(EmployeeId) |

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

Листинг 1 – SQL-запрос для создания таблицы SupportCase

--Создание таблицы SupportCase с полями и первичным ключом

CREATE TABLE [dbo].[SupportCase](

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

[SystemID] [int] NULL,

[ReportedByEmpID] [int] NULL,

[CaseDetails] [nvarchar](max) NULL,

[CaseStatus] [nvarchar](50) NULL,

[OpenedOn] [datetime] NULL,

[ClosedOn] [datetime] NULL,

PRIMARY KEY CLUSTERED

(

[CaseID] 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 уникальные 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 на название приложения

GO

ALTER TABLE Application

ADD CONSTRAINT UQ\_Application\_AppTitle UNIQUE (AppTitle);

GO

Для решения аналитических задач в системе создаётся скалярная функция GetCaseCountBySystem, возвращающая количество обращений, связанных с конкретным устройством (системой). Код создания функции приведён в **листинге 3.**

Листинг 3 – Код создания скалярной функции GetCaseCountBySystem

-- Функция возвращает количество обращений по конкретному устройству

CREATE FUNCTION GetCaseCountBySystem

(

@SystemID INT

)

RETURNS INT

AS

BEGIN

DECLARE @CaseCount INT;

SELECT @CaseCount = COUNT(\*)

FROM SupportCase

WHERE SystemID = @SystemID;

RETURN @CaseCount;

END

Для отображения сведений об установленных приложениях с привязкой к устройствам и сотрудникам создаётся представление AppWithSystemAndEmployee. Код создания приведён в **листинге 4**.

Листинг 4 – Код создания представления AppWithSystemAndEmployee

-- Представление показывает приложения, устройства и сотрудников

CREATE VIEW AppWithSystemAndEmployee AS

SELECT

a.AppTitle AS Application,

a.AppEdition,

a.SerialCode,

m.SystemName,

m.NetworkIP,

e.FullName AS AssignedEmployee

FROM Application a

LEFT JOIN Machine m ON a.SystemID = m.SystemID

LEFT JOIN Employees e ON m.AssignedEmpID = e.EmpID;

Для обеспечения контроля над заполнением обращений реализуется триггер trg\_BlockEmptyCaseDetails, запрещающий добавление записей с пустыми описаниями. Код приведён в **листинге 5**.

Листинг 5 – Код создания триггера trg\_BlockEmptyCaseDetails

-- Триггер блокирует вставку обращений без описания

CREATE TRIGGER trg\_BlockEmptyCaseDetails

ON SupportCase

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

IF EXISTS (

SELECT 1

FROM inserted

WHERE LTRIM(RTRIM(ISNULL(CaseDetails, ''))) = '')

BEGIN

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

ROLLBACK TRANSACTION;

RETURN;

END

END

Одним из этапов настройки системы является заполнение ключевых таблиц начальными данными. В рамках этого этапа был выполнен ручной ввод информации об устройствах с помощью базовой операции INSERT INTO. Код добавления записи в таблицу Machine приведён в **листинге 6**.

Листинг 6 – Код для добавления устройства в таблицу Machine

-- Добавление записи об устройстве в таблицу Machine

INSERT INTO Machine (

SystemID,

SystemName,

NetworkIP,

AdapterMAC,

HardwareType,

OperatingSystem,

Site,

AssignedEmpID

)

VALUES (

1,

'WKS-STORAGE01',

'192.168.0.101',

'00-1A-2B-3C-4D-5E',

'Складской терминал',

'Windows 10 Pro',

'Главный склад',

2

);

## 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 Application преобразуются и сохраняются в поле LicenseHash с помощью SQL-функции HASHBYTES. При этом:

* значение приводится к типу VARBINARY;
* 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, реализующего хеширование серийных ключей, приведён в **листинге 7**.

Листинг 7 – Код хеширования ключей в таблице Application

-- Обновление таблицы Application с хешированием лицензионных ключей

UPDATE Application

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

WHERE SerialCode IS NOT NULL;

Для ограничения доступа к данным создаются отдельные логины и пользователи базы данных. Сотрудникам предоставляются права только на чтение, в то время как техническим специалистам разрешены действия по модификации. Соответствующий SQL-код показан в **листинге 8**.

Листинг 8 – Код создания логина и пользователя с правом чтения

-- Создание логина и пользователя с правом только на чтение

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

CREATE USER User\_ReadOnly FOR LOGIN User\_ReadOnly;

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

Для защиты данных от потери необходимо регулярно выполнять **резервное копирование базы данных**. Это позволяет восстановить информацию после сбоя, повреждения или непредвиденной ситуации. Пример SQL-скрипта для создания полной резервной копии базы данных приведён в **листинге 9**.

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

-- Выполнение резервного копирования БД

BACKUP DATABASE [IT\_Inventory] TO

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

WITH NOFORMAT, NOINIT,

NAME = N'Полная резервная копия IT\_Inventory',

SKIP, NOREWIND, NOUNLOAD, STATS = 10;

Одним из ключевых этапов стратегии защиты данных является возможность **восстановления базы из резервной копии**. Это позволяет в кратчайшие сроки вернуть систему к рабочему состоянию. Пример восстановления базы данных из ранее созданной резервной копии представлен 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 = 1, NOUNLOAD, STATS = 5;

# Выполнение работ по ПМ.01

## Проектирование ПО

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

* учёт поступлений и отгрузок материалов;
* отображение текущих остатков по складу;
* логирование действий;
* регистрация и поиск номенклатуры.

Разрабатываемое программное обеспечение предназначено для использования сотрудниками склада и материально-ответственных лиц. Решение позволяет исключить ошибки ручного учёта, обеспечить быстрый доступ к данным и автоматизировать процессы инвентаризации и контроля.

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

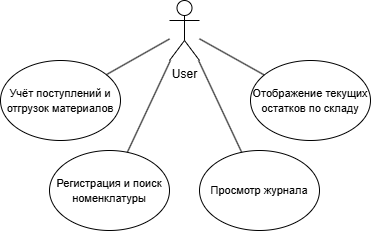


Рисунок 4 – Диаграмма прецедентов

## Разработка программных модулей

В качестве языка разработки использовался C# [1] с использованием технологии WPF для создания интерактивного графического интерфейса пользователя. Одним из основных компонентов системы является **обработка действий пользователя с ячейками склада** – в частности, перемещение, добавление и редактирование материалов. Реализация логики перемещения между ячейками осуществляется в обработчиках событий DragDrop, связанных с визуальными элементами склада. Соответствующий фрагмент кода приведён в **листинге 11**.

Листинг 11 – Код обработчика событий DragDrop

// Код обработчика событий DragDrop.

private void Cell\_Drop(object sender, DragEventArgs e)

{

FrameworkElement element = sender as FrameworkElement;

if (element == null) return;

WarehouseCell cell = element.DataContext as WarehouseCell;

if (cell == null) return;

if (e.Data.GetDataPresent(typeof(Material)))

{

Material material = e.Data.GetData(typeof(Material)) as Material;

if (material == null) return;

var viewModel = (WarehouseViewModel)DataContext;

viewModel.SelectedMaterial = material;

viewModel.DropOnCellCommand.Execute(cell);

}

else if (e.Data.GetDataPresent(typeof(WarehouseCell)))

{

WarehouseCell sourceCell = e.Data.GetData(typeof(WarehouseCell)) as WarehouseCell;

if (sourceCell == null) return;

var viewModel = (WarehouseViewModel)DataContext;

viewModel.MoveMaterialCommand.Execute((sourceCell, cell));

}

}

Используя средства WPF [5], требуется спроектировать интерфейс для управления складскими ячейками в соответствии с рисунком 5.

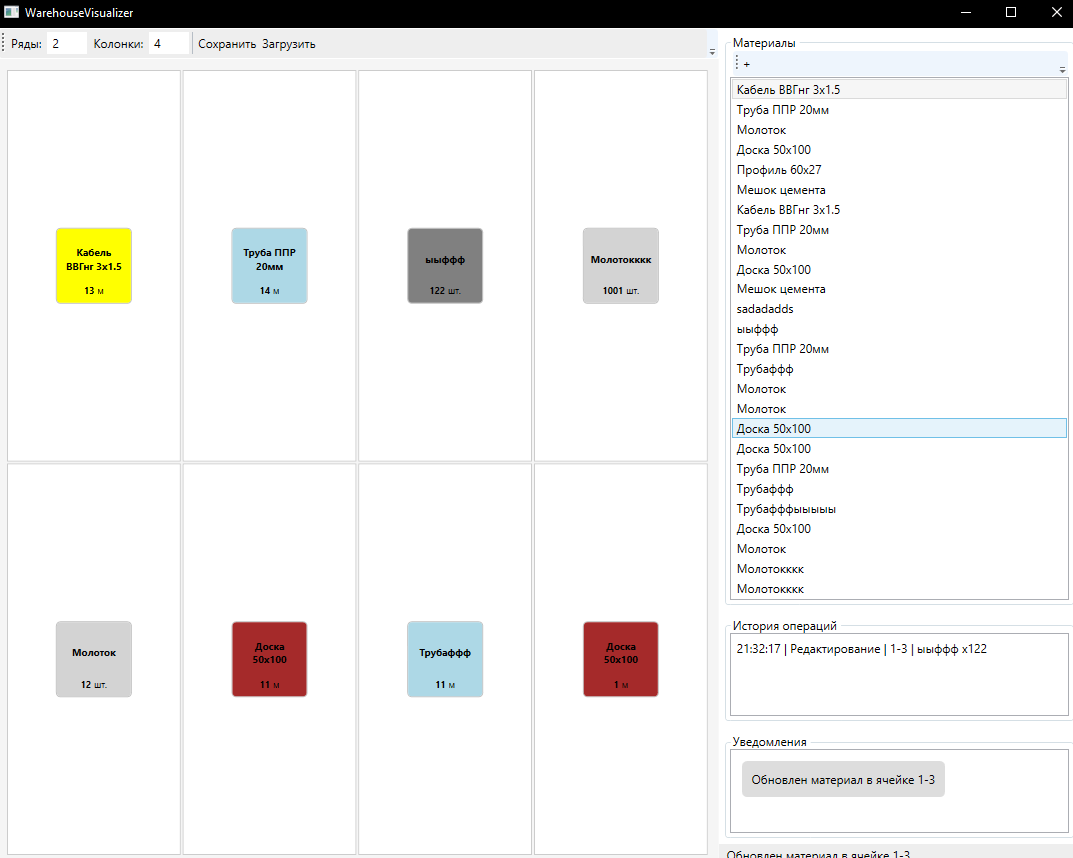


Рисунок 5 – WarehouseVisualizer. Вид основного окна

## Разработка мобильного приложения

Предприятием была поставлена задача разработать простое и удобное мобильное приложение для создания, хранения и просмотра личных текстовых заметок.

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 [4].

Главная страница приложения NotesApp (рисунок 6) cодержит поле для ввода текста и кнопка сохранения.

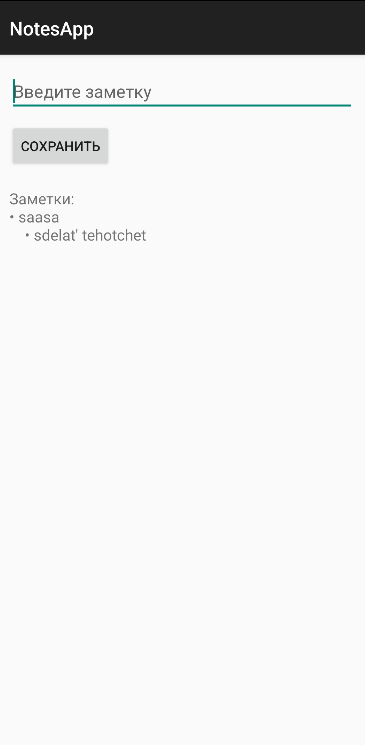


Рисунок 6 – NotesScreen. Вид главной страницы

Пользователь может ввести текст заметки и нажать кнопку «Сохранить». После этого данные сохраняются в локальное хранилище и сразу отображаются в нижней части экрана. Логика обработки событий и сохранения данных реализована в MainActivity. Код представлен в **листинге 12**.

Листинг 12 – Логика заполнения и обработки кнопки

class MainActivity : AppCompatActivity() {  
  
 private lateinit var editTextNote: EditText  
 private lateinit var buttonSave: Button  
 private lateinit var textViewNotes: TextView  
  
 private val notesKey = "notes"  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 editTextNote = findViewById(R.id.*editTextNote*)  
 buttonSave = findViewById(R.id.*buttonSave*)  
 textViewNotes = findViewById(R.id.*textViewNotes*)  
  
 val sharedPref = getSharedPreferences("MyNotes", Context.*MODE\_PRIVATE*)  
 val savedNotes = sharedPref.getString(notesKey, "")  
 textViewNotes.*text* = "Заметки:\n$savedNotes"  
  
 buttonSave.setOnClickListener {val newNote = editTextNote.*text*.toString()  
 if (newNote.*isNotBlank*()) {  
 val currentNotes = sharedPref.getString(notesKey, "")  
 val updatedNotes = "$currentNotes• $newNote\n"  
  
 *with*(sharedPref.edit()) {putString(notesKey, updatedNotes)  
 apply()  
 }textViewNotes.*text* = "Заметки:\n$updatedNotes"  
 editTextNote.*text*.clear()  
 }  
 }}  
}

## Отладка и тестирование программных модулей

Для обеспечения стабильной и надёжной работы приложения 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** |
| Переместить материал из списка в свободную ячейку | Материал добавляется в ячейку, отображается визуально | Соответствует ожидаемому |
| Перетащить материал из одной ячейки в другую | Материал перемещается, история действия обновляется | Соответствует ожидаемому |
| Очистить заполненную ячейку | Ячейка становится пустой, история пополняется записью «Удаление» | Соответствует ожидаемому |
| Сохранить склад в БД | Данные сохраняются, появляется уведомление об успешной операции | Соответствует ожидаемому |

Также выполнено **автоматизированное тестирование** с использованием фреймворка xUnit. Разработан модульный тест для проверки корректности работы метода Clone() класса Material. Результаты проверены вручную и успешно пройдены. Код представлен в **листинге 13**.

Листинг 13 – Код unit-тест Clone\_ShouldCreateExactCopy()

[Fact]

public void Clone\_ShouldCreateExactCopy()

{

// Arrange

var original = new Material

{

Name = "Steel Beam",

Quantity = 50,

Type = MaterialType.Metal,

Unit = "kg"

};

// Act

var copy = (Material)original.Clone();

// Assert

Assert.Equal(original.Name, copy.Name);

Assert.Equal(original.Quantity, copy.Quantity);

Assert.Equal(original.Type, copy.Type);

Assert.Equal(original.Unit, copy.Unit);

Assert.NotSame(original, copy);

}

## Оптимизация и рефакторинг программного кода

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

Проведённые улучшения включают:

* 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. В листинге 14 приведен фрагмент кода с исправлением именования.

Листинг 14 – Фрагмент кода с изменениями

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

// После рефакторинга:

var newMaterial = new Material();

Дополнительно была оптимизирована загрузка склада с базы данных. Для обеспечения отзывчивости интерфейса метод загрузки был реализован асинхронно (листинг 15).

Листинг 15 – Асинхронный запуск загрузки склада

// До:

var warehouse = \_sqlDataService.LoadWarehouse();

// После:

var warehouse= await Task.Run(() => \_sqlDataService.LoadWarehouse());

Заключение

Для прохождения практики в строительной компании ООО «ВЕГА-С» 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:

* сформированы алгоритмы разработки ПМ в соответствии с техническим заданием;
* разработан ПМ в соответствии с техническим заданием, с использованием современных языков программирования и сред разработки;
* выполнены тестирования ПМ;
* осуществлен рефакторинг и оптимизация программного кода;
* разработаны модули ПО для мобильных платформ;
* осуществлены сбор, обработка и анализ информации для проектирования БД;
* спроектированы базы данных в соответствии с результатами анализа предметной области, с учётом требований к структуре, целостности и производительности;
* реализована БД в конкретной СУБД;
* произведено администрирование БД;
* обеспечена защита информации в базах данных с использованием современных технологий и методов информационной безопасности.

Список использованных источников

1. 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). – Режим доступа: свободный. – Текст: электронный.
2. Пахомов Б. И. C#для начинающих. – Санкт-Петербург : БХВ-Петербург, 2014. – 432 с. – URL: <https://lesmatveev.narod.ru/knigi_proect/ci_sharp.pdf> (дата обращения: 21.05.2025). – Режим доступа: свободный. – Текст: электронный.
3. Шустова, Л. И. Базы данных: учебник / Л. И. Шустова, О. В. Тараканов. – Москва : ИНФРА-М, 2024. – 304 с. – Текст : электронный. – URL: <https://znanium.ru/catalog/product/2149043> (дата обращения: 29.05.2025). – Режим доступа: по подписке.
4. JetBrains. Документацияпо Kotlin для Android-разработчиков. – URL: <https://kotlinlang.org/docs/android-overview.html> (дата обращения: 12.05.2025). – Режим доступа: свободный. – Текст: электронный.
5. 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: 05.05.2025). – Текст : электронный.