МИНИСТЕРСТВО ОБРАЗОВАНИЯ КИРОВСКОЙ ОБЛАСТИ

Кировское областное государственное профессиональное образовательное

бюджетное учреждение

«Слободской колледж педагогики и социальных отношений»

**ОТЧЕТ**

**по производственной практике**

**ПМ.02. Осуществление интеграции программных модулей**

Студент

Коротких Наталья Михайловна

Группа 21П-1

Специальность 09.02.07 Информационные системы и программирование

Руководитель практики от колледжа:

*Пентин Николай Сергеевич*

Руководитель практики от организации:

*\_\_\_\_\_\_\_\_/Калинин Арсений Олегович*

подпись

УТВЕРЖДАЮ:

Директор

КОГПОБУ Слободской колледж педагогики и социальных отношений   
 Наименование организации

\_\_\_\_\_\_\_\_\_\_\_/ Шеренцова О. М.\_\_\_\_\_

подпись расшифровка

М. П.

2023-2024 уч. год

Оглавление

[**1.** **ХАРАКТЕРИСТИКА ОБЪЕКТА ПРАКТИКИ (ЮРИДИЧЕСКИЙ АДРЕС, СПЕЦИАЛИЗАЦИЯ, СТРУКТУРА)** 3](#_Toc170927165)

[**2.** **ОПИСАНИЕ РАБОЧЕГО МЕСТА** 4](#_Toc170927166)

[**3.** **СОСТАВ ПРОГРАММНОГО И ТЕХНИЧЕСКОГО ОБЕСПЕЧЕНИЯ, ИМЕЮЩЕГОСЯ НА ПРЕДПРИЯТИИ, ИХ НАЗНАЧЕНИЕ** 5](#_Toc170927167)

[**4.** **ОПИСАНИЕ ВЫПОЛНЕННЫХ ВИДОВ РАБОТ** 6](#_Toc170927168)

[**4.1.** **Разработан и обоснован вариант интеграционного решения с помощью графических средств среды разработки** 6](#_Toc170927169)

[**4.2.** **Результаты верно сохранены в системе контроля версий** 10](#_Toc170927170)

[**4.3.** **В системе контроля версий выбрана верная версия проекта. Проанализирована его архитектура** 11](#_Toc170927171)

[**4.4.** **Архитектура доработана для интеграции нового модуля** 12](#_Toc170927172)

[**4.5.** **Выбраны способы форматирования данных и организована их постобработка, транспортные протоколы и форматы сообщений обновлены** 13](#_Toc170927173)

[**4.6.** **Выполнена доработка модуля и дополнительная обработка исключительных ситуаций, в том числе, с созданием классов-исключений** 15](#_Toc170927174)

[**4.7.** **Определены качественные показатели полученного проекта** 17](#_Toc170927175)

[**4.8.** **Результат интеграции сохранен в системе контроля версий** 19](#_Toc170927176)

[**4.9.** **Протестирована интеграция модулей проекта и выполнена отладка проекта с применением инструментальных средств среды** 20](#_Toc170927177)

[**4.10.** **Разработан тестовый сценарий и тестовые пакеты** 22](#_Toc170927178)

[**4.11.** **Выполнено тестирование интеграции и ручное тестирование** 25](#_Toc170927179)

[**4.12.** **Выявлены ошибки системных компонентов (при наличии)** 35](#_Toc170927180)

[**4.13.** **Заполнены протоколы тестирования** 36](#_Toc170927181)

[**4.14.** **Продемонстрировано знание стандартов кодирования более чем одного языка программирования** 38](#_Toc170927182)

[**4.15.** **Выявлены все имеющиеся несоответствия стандартам в предложенном коде** 39](#_Toc170927183)

[**5.** **РУКОВОДСТВО ОПЕРАТОРА** 41](#_Toc170927184)

[**6.** **ЗАКЛЮЧЕНИЕ** 44](#_Toc170927185)

[**7.** **ПРИЛОЖЕНИЯ К ОТЧЕТУ: ОТЧЕТ В ЭЛЕКТРОННОМ ВИДЕ, ПРЕЗЕНТАЦИЯ ДЛЯ ВЫСТУПЛЕНИЯ И ДР. МАТЕРИАЛЫ** 46](#_Toc170927186)

# **ХАРАКТЕРИСТИКА ОБЪЕКТА ПРАКТИКИ (ЮРИДИЧЕСКИЙ АДРЕС, СПЕЦИАЛИЗАЦИЯ, СТРУКТУРА)**

Наименование базы практики – Кировское областное государственное профессиональное образовательное бюджетное учреждение «Слободской колледж педагогики и социальных отношений» (КОГПОБУ СКПиСО).

Адрес – Кировская область, г. Слободской, ул. Рождественская, д. 69.

ФИО руководителя (директора) – Шеренцова Ольга Михайловна.

Телефон директора +79014791707.

Образование в КОГПОБУ СКПиСО ориентировано на массовое обучение специалистов с рабочими навыками. Половина учебного времени отводится на практические работы в лабораториях. Это помогает выпускникам органично и быстро включаться в трудовые процессы на новых местах работы.

Схема организационной структуры представлена на рисунке 1.

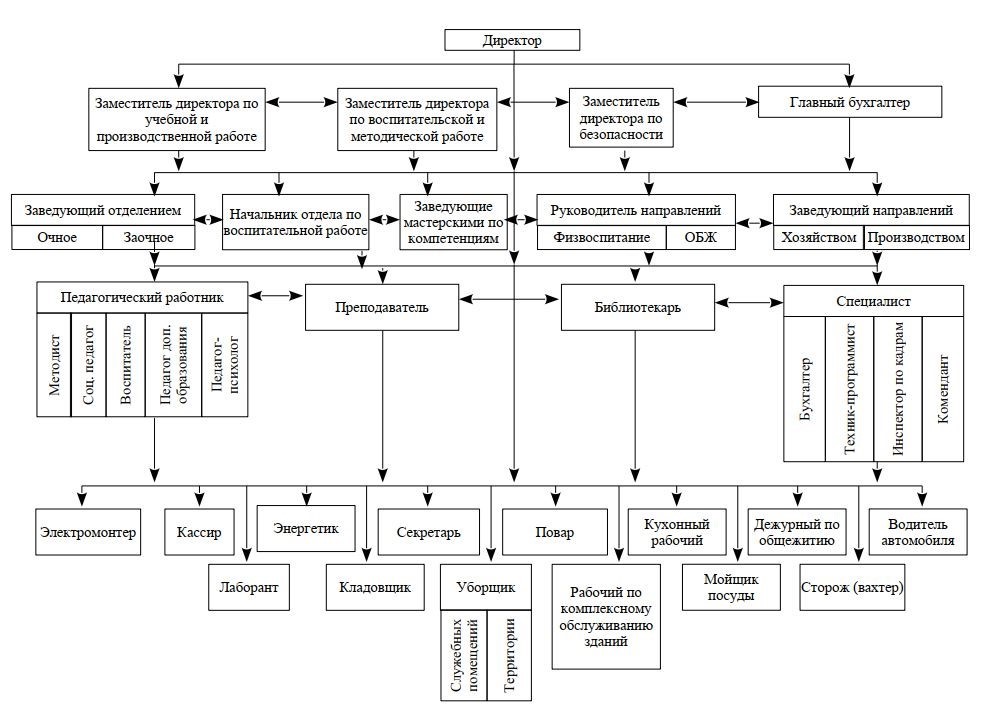


Рисунок 1 – Схема организационной структуры

# **ОПИСАНИЕ РАБОЧЕГО МЕСТА**

Рабочее место оснащено ноутбуком с операционной системой Майкрософт Windows 10 Pro,процессор **–** AMD 3020e with Radeon Graphics, 1200 МГц, ядер: 2, логических процессоров: 2.

Для работы потребовались - SQL Server Management Studio (SSMS), Visual Studio Community 2022, Microsoft Word, Microsoft Visio.

Фото рабочего места представлено на рисунке 2.

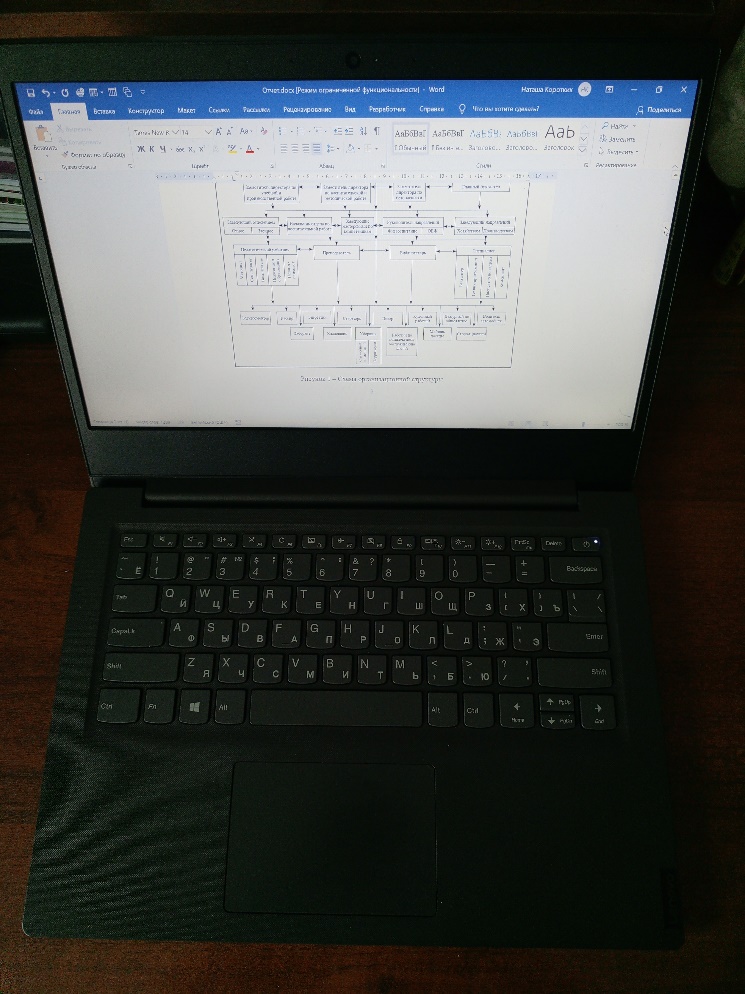


Рисунок 2 – Рабочее место

# **СОСТАВ ПРОГРАММНОГО И ТЕХНИЧЕСКОГО ОБЕСПЕЧЕНИЯ, ИМЕЮЩЕГОСЯ НА ПРЕДПРИЯТИИ, ИХ НАЗНАЧЕНИЕ**

На предприятии имеется специализированное программное обеспечение, предназначенное для работы с учебными материалами, проведения онлайн-обучения и тестирования студентов на время учебного процесса.

Техническое обеспечение колледжа включает в себя компьютеры и ноутбуки с операционной системой Windows 7, 10, принтеры, интерактивные доски, клавиатуры и мыши, необходимые для проведения занятий и выполнения учебных заданий. В одном из кабинетов присутствует интерактивная система голосования, в другом кабинете оборудован конференц-зал. Компьютеры и устройства в учебных кабинетах объединены в единую локальную сеть, имеют системы защиты от вирусов и несанкционированного доступа, повреждения данных.

Программное обеспечение на предприятии включает в себя специализированные программы для работы с различными типами файлов (Microsoft Word, PowerPoint, Excel, Visio), создания презентаций (Microsoft PowerPoint) и хранение, обработки данных (1С: Предприятие (учебная версия), Visual Studio Community 2022, SQL Server Management Studio).

В целом, на предприятии имеется все необходимое техническое и программное обеспечение для успешного проведения образовательного процесса и обучения студентов.

# **ОПИСАНИЕ ВЫПОЛНЕННЫХ ВИДОВ РАБОТ**

# **Разработан и обоснован вариант интеграционного решения с помощью графических средств среды разработки**

Для данной практики я дорабатываю базу данных и приложение, которые были разработаны для производственной практики по модулю ПМ.11 Разработка, администрирование и защита баз данных.

После изучения предметной области для будущей базы данных была построена ER – диаграмма в Microsoft Visio 2019 (Рисунок 4.1.1).

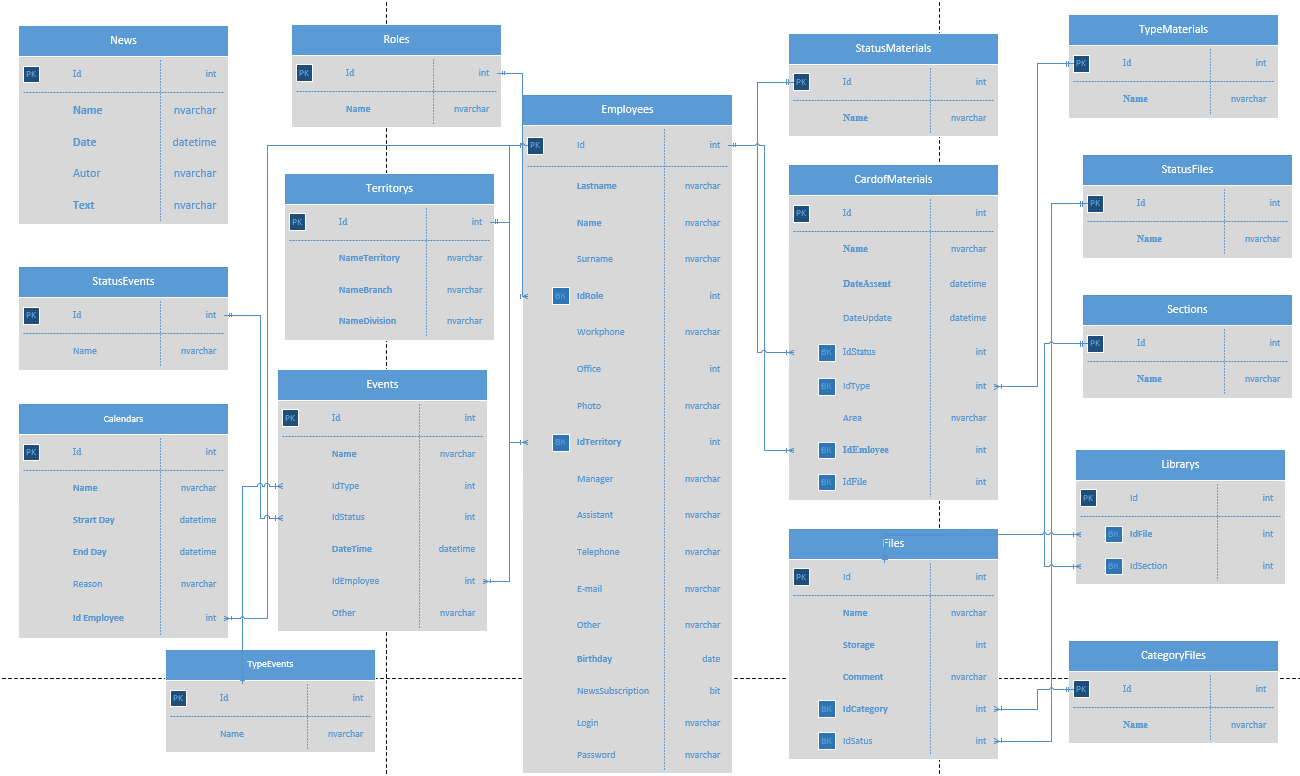


Рисунок 4.1.1 – ER-диаграмма базы данных

Для создания и настройки базы данных для внутреннего корпоративного портала ГК «Дороги России», которая разрабатывалась в рамках подготовки к отборочному чемпионату по компетенции «Программные решения для бизнеса» я использовала SQL Server Management Studio (SSMS).

Были созданы таблицы, триггеры и другие объекты базы данных, необходимые для работы интеграционного решения. Схема базы данных представлена на рисунке 4.1.2.

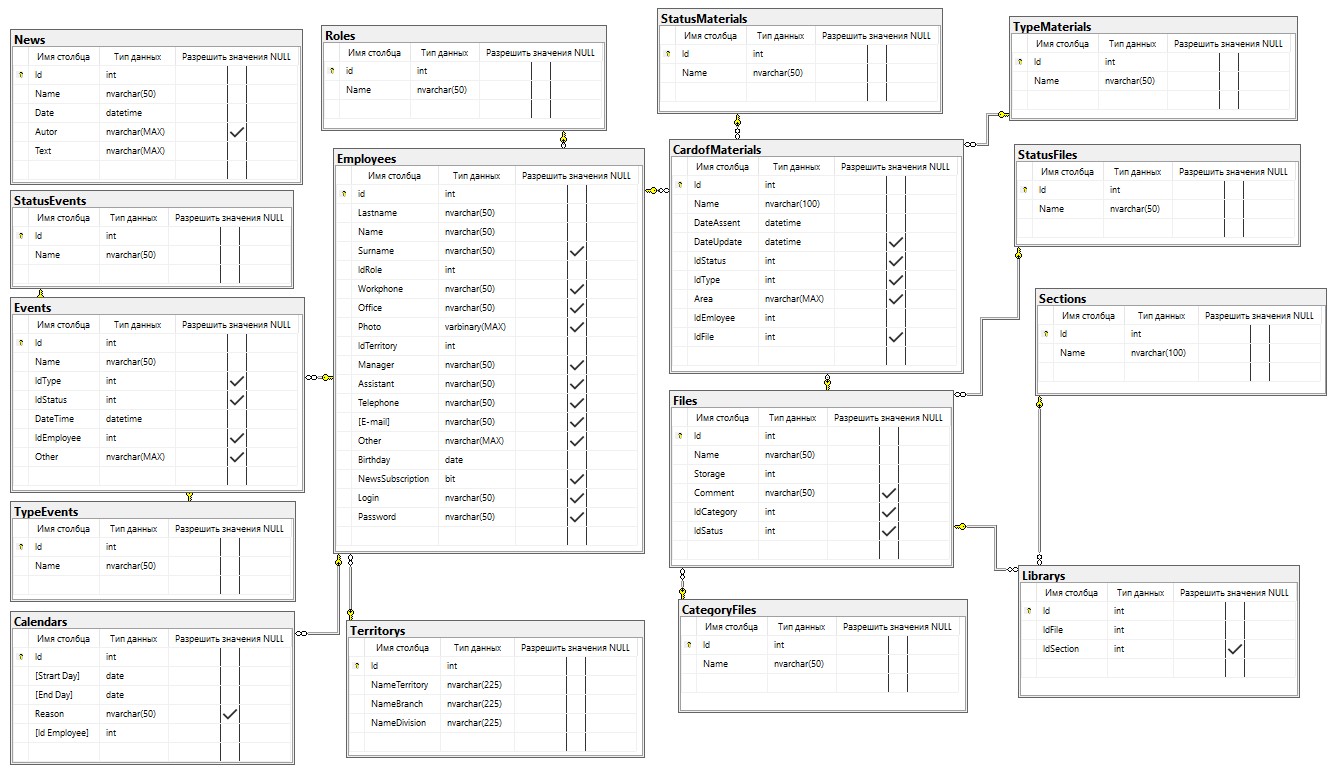


Рисунок 4.1.2 – Схема БД

Потом было выполнено создание проекта в среде разработки Microsoft Visual Studio. В проекте были созданы необходимые классы и методы для работы с данными, а также было настроено подключение к базе данных (Рисунок 4.1.3). Это позволило легко и эффективно управлять данными в процессе работы.

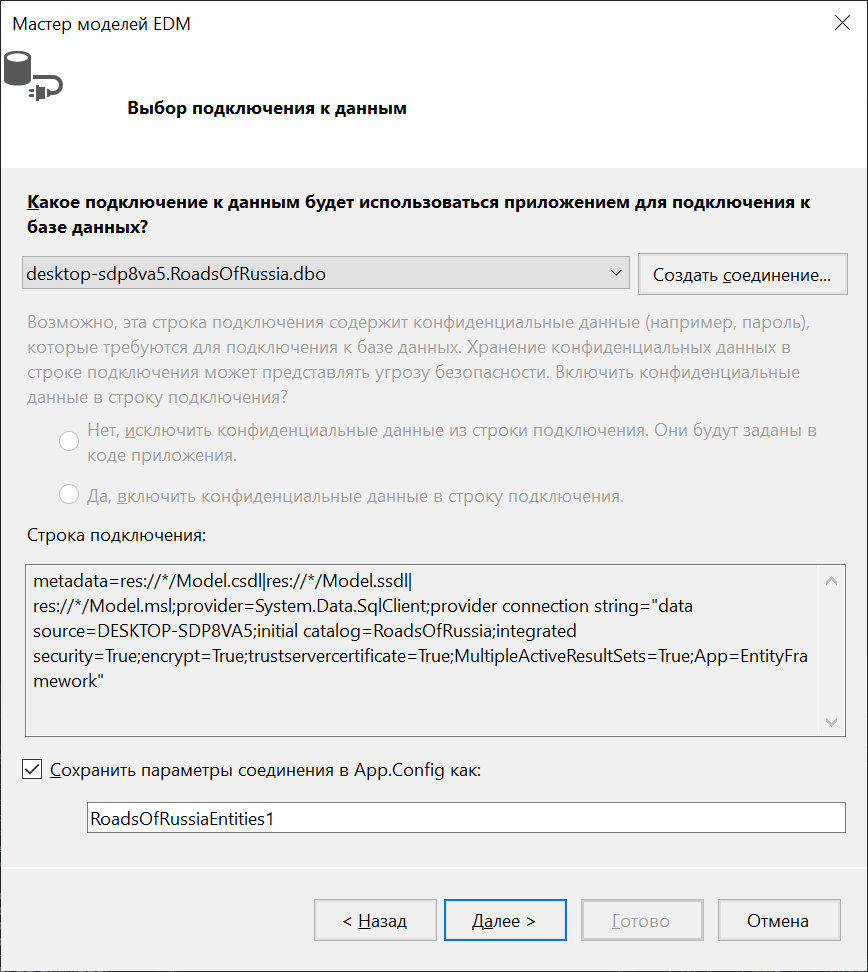


Рисунок 4.1.3 – Подключение к БД

Разработанный мной проект содержал окно авторизации (Рисунок 4.1.4).

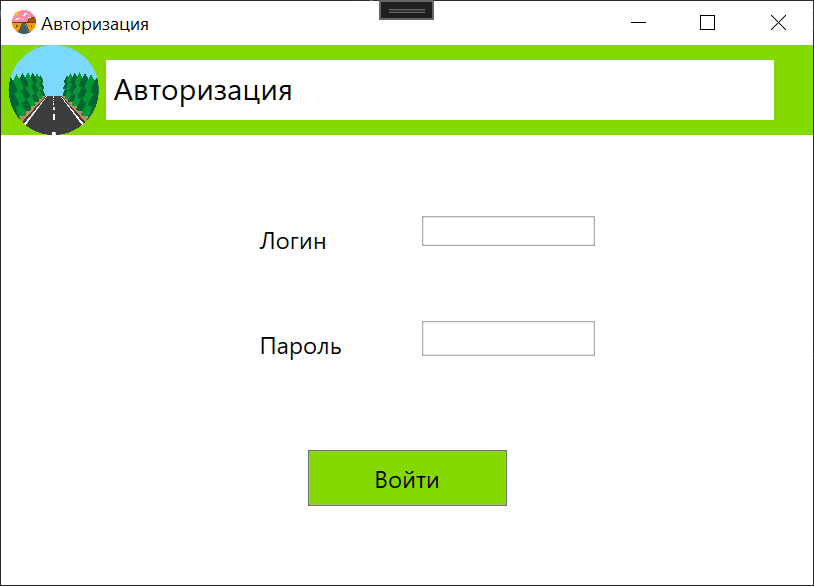


Рисунок 4.1.4 – Окно авторизации

Окно со списком всех сотрудников организации (Рисунок 4.1.5).

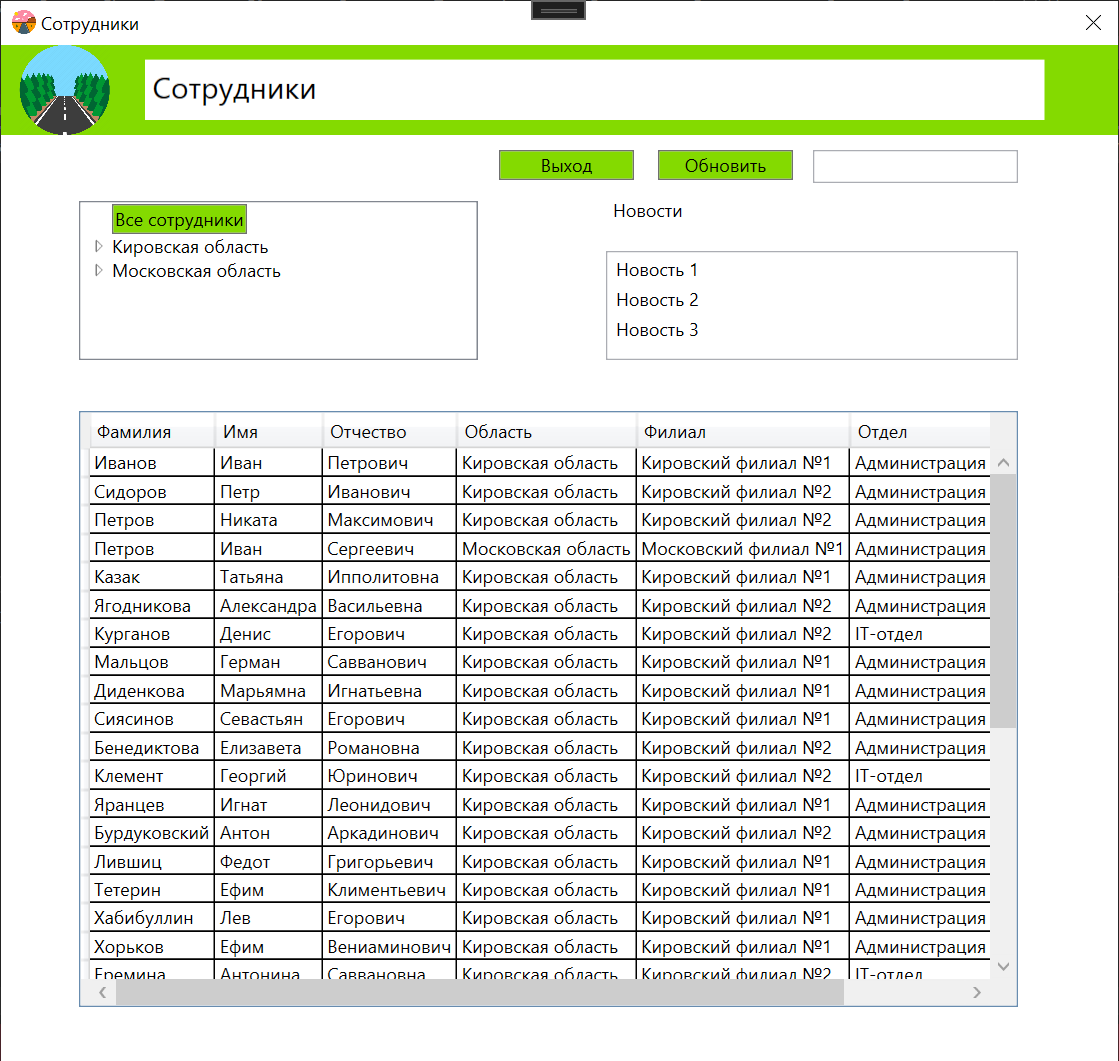


Рисунок 4.1.5 – Окно со списком всех сотрудников

Функциональные возможности: при выборе области или филиала, выводится список тех сотрудников, которые работаю в этой области и филиале (Рисунок 4.1.6).

На главном окне также присутствует поиск по ФИО, должности и почте. Результаты поиска по имени - Антон представлены на рисунке 4.1.7.

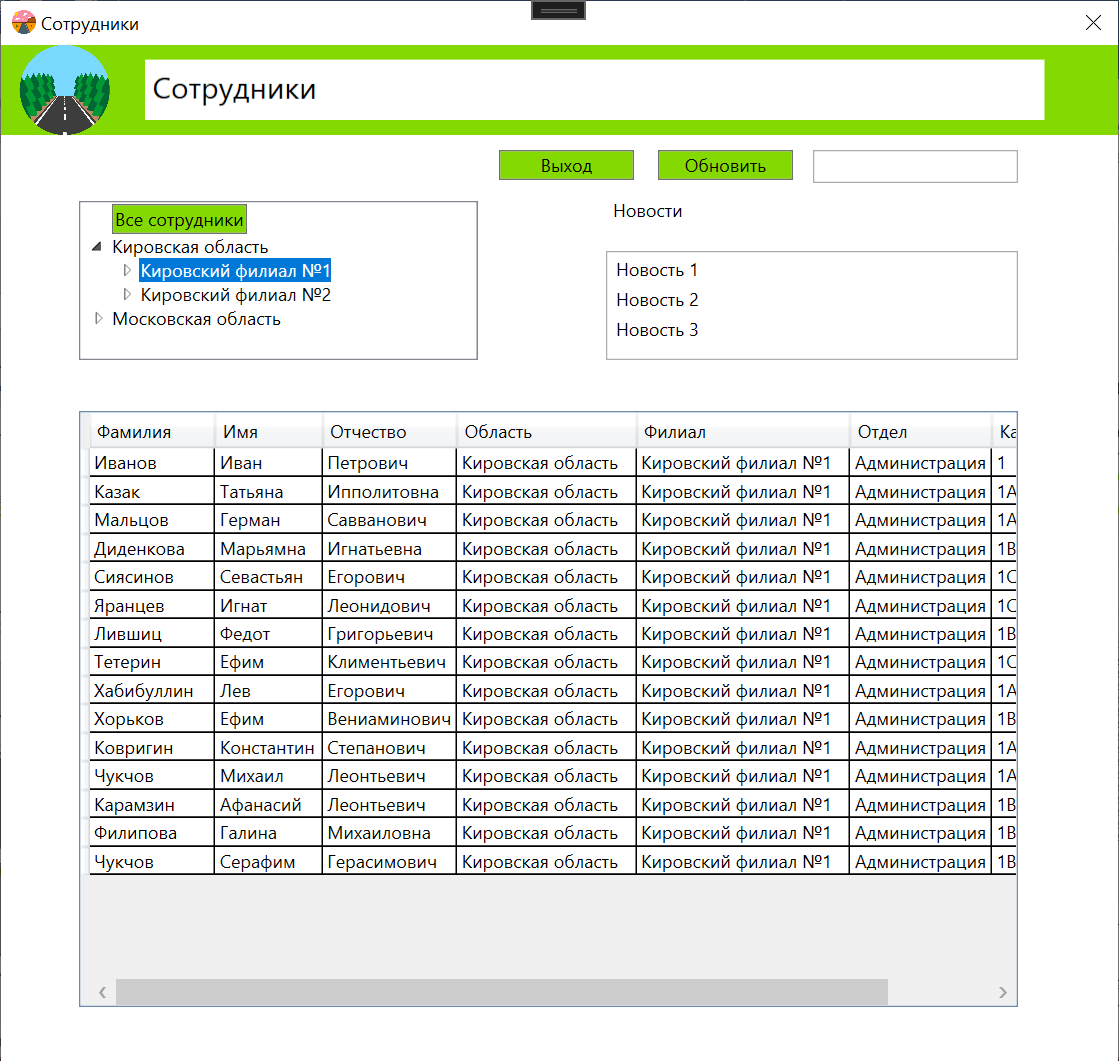


Рисунок 4.1.6 – Результат фильтрации

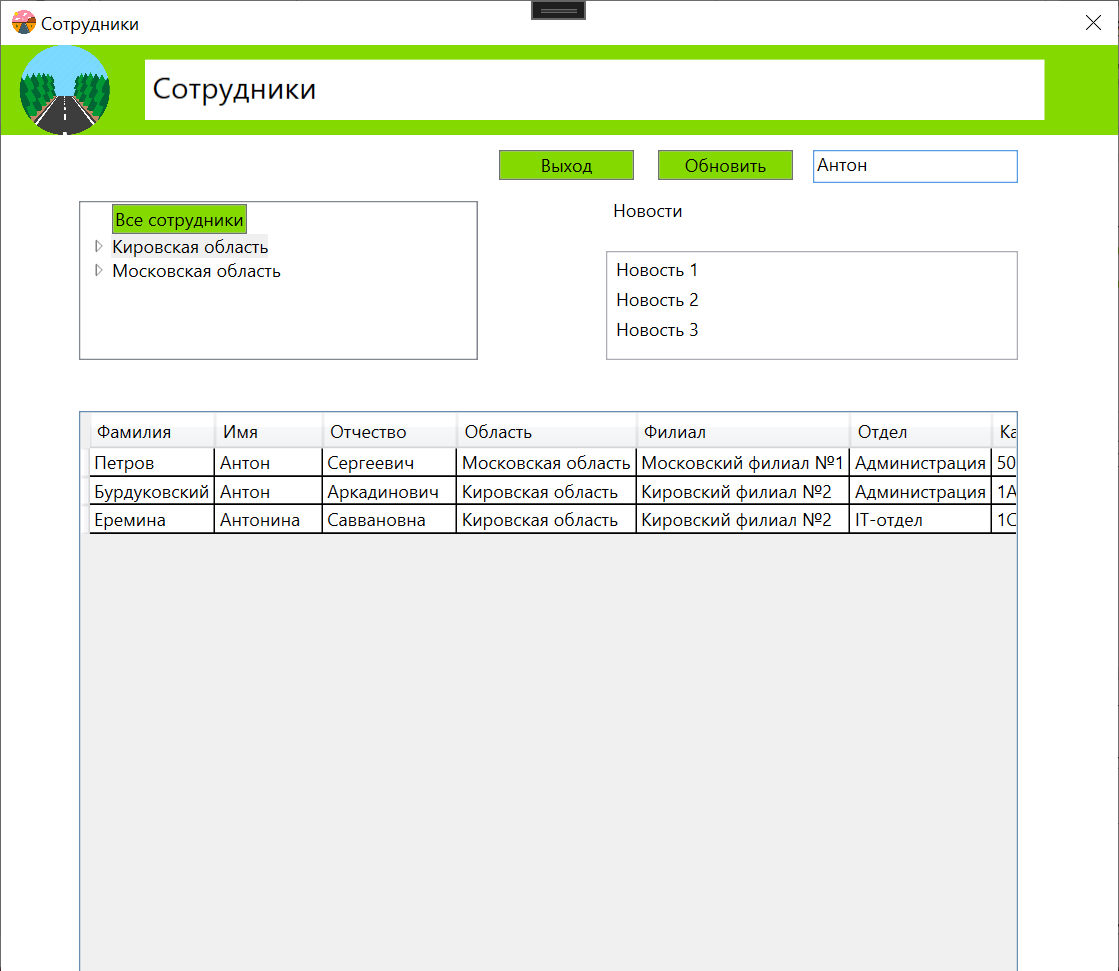


Рисунок 4.1.7 – Результат поиска

# **Результаты верно сохранены в системе контроля версий**

Для обеспечения версионного контроля и возможности отслеживания изменений в коде проекта была использована система контроля версий Git, а в качестве хостинга - платформа GitHub.

*Создание репозитория и загрузка проекта:*

Был создан репозиторий на GitHub с названием «Industrial practice PM.02» (ссылка - <https://github.com/Natasha975/Industrial-practice-PM.02.git>). Для загрузки проекта в репозиторий использовался клиент Git - GitHub Desktop (Рисунок 4.2.1).

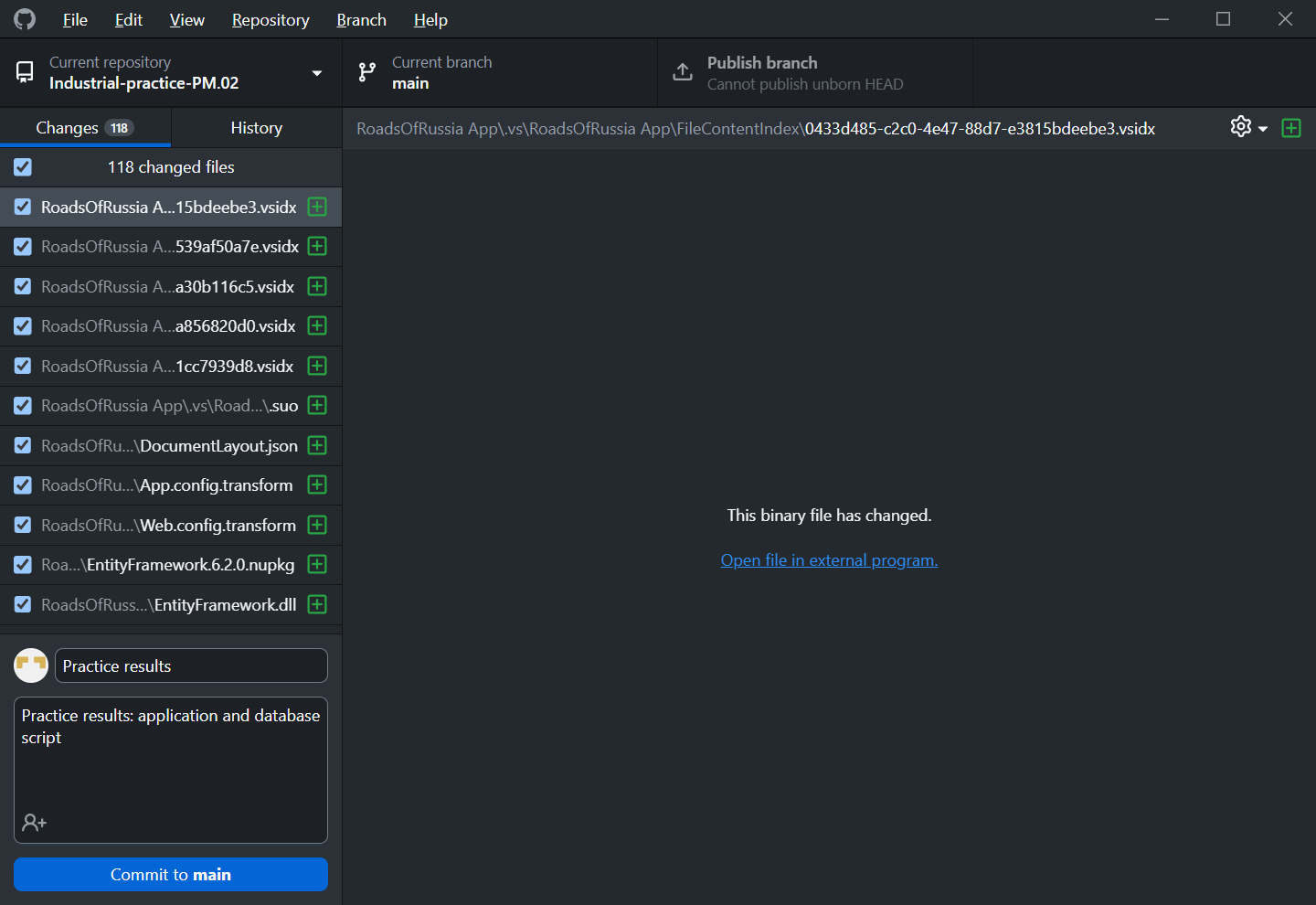


Рисунок 4.2.1 – Сохранение результатов в системе контроля версий

*Комментирование кода:*

Для повышения читаемости и понимания кода проекта был добавлен комментарий. Комментарии содержат информацию о содержании проекта.

*История изменений:*

В репозитории доступна полная история изменений проекта, включая все коммиты, которые содержат описание внесенных изменений. Это позволяет отслеживать развитие проекта и восстанавливать предыдущие версии кода в случае необходимости.

*Безопасность:*

Система контроля версий Git обеспечивает надежную защиту от потери данных и позволяет отслеживать все изменения в коде.

Использование системы контроля версий Git и GitHub позволило обеспечить версионный контроль проекта, а также упростило работу над проектом, повысило его безопасность.

# **В системе контроля версий выбрана верная версия проекта. Проанализирована его архитектура**

Т. к. разработанное приложение связано с базой данной, то на уровне доступа к источнику данных приложение использует объектно-ориентированную технологию доступа к данным – ADO.NET Entity Framework (Рисунок 4.3.1).

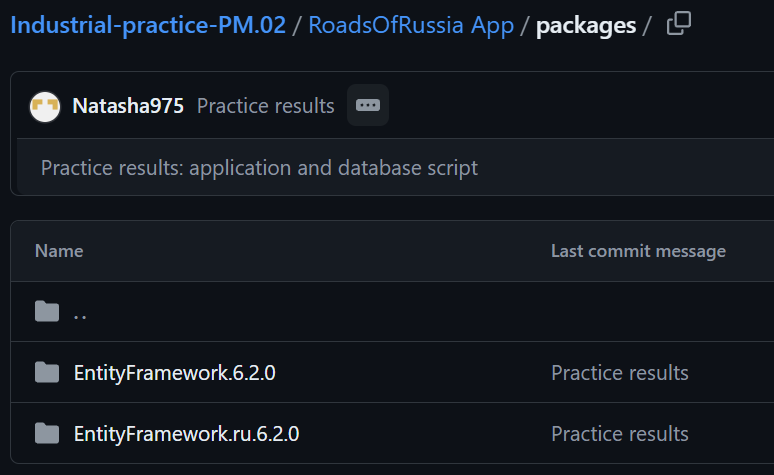


Рисунок 4.3.1 - ADO.NET Entity Framework

Модуль обработки данных расположенного на уровне бизнес логики заключается в обработке данных, полученных из СУБД в формат данных, который будет понятен пользователю. Уровень доступа к приложению, содержащий модуль интерфейса пользователя включает окна с отображением данных, функций и настроек доступных для пользователя. Все вместе модули описывают структуру приложения, готового для использования конечным пользователем (Рисунок 4.3.2).

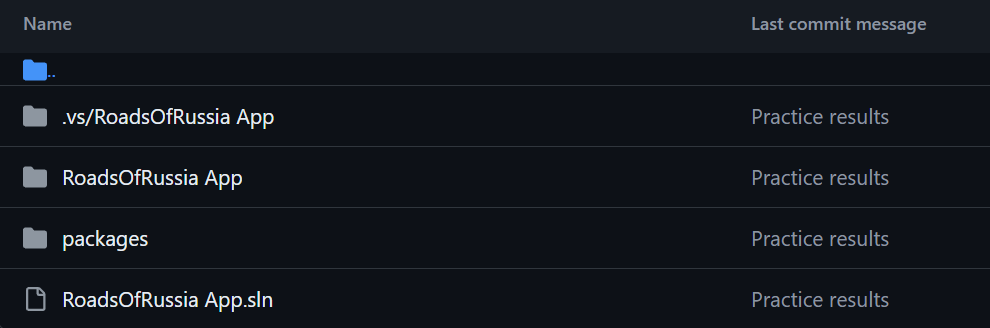


Рисунок 4.3.2 – Структура приложения

# **Архитектура доработана для интеграции нового модуля**

Для успешной интеграции нового модуля в проект была проведена доработка архитектуры. В результате были произведены следующие изменения:

*Создание нового модуля:*

Был создан отдельный модуль «Обработка ошибок», отвечающий за централизованное управление ошибками в приложении.

*Расширение существующих модулей:*

* Модуль обработки данных:

Были добавлены функции для взаимодействия с новым модулем обработки ошибок, позволяющие регистрировать и обрабатывать ошибки на уровне данных.

Внесены изменения в логику обработки данных, чтобы она корректно взаимодействовала с новым модулем обработки ошибок.

* Модуль интерфейса пользователя:

Добавлены элементы пользовательского интерфейса, такие как окна (Рисунок 4.4.1), кнопки, сообщения, которые позволяют пользователю взаимодействовать с модулем обработки ошибок.

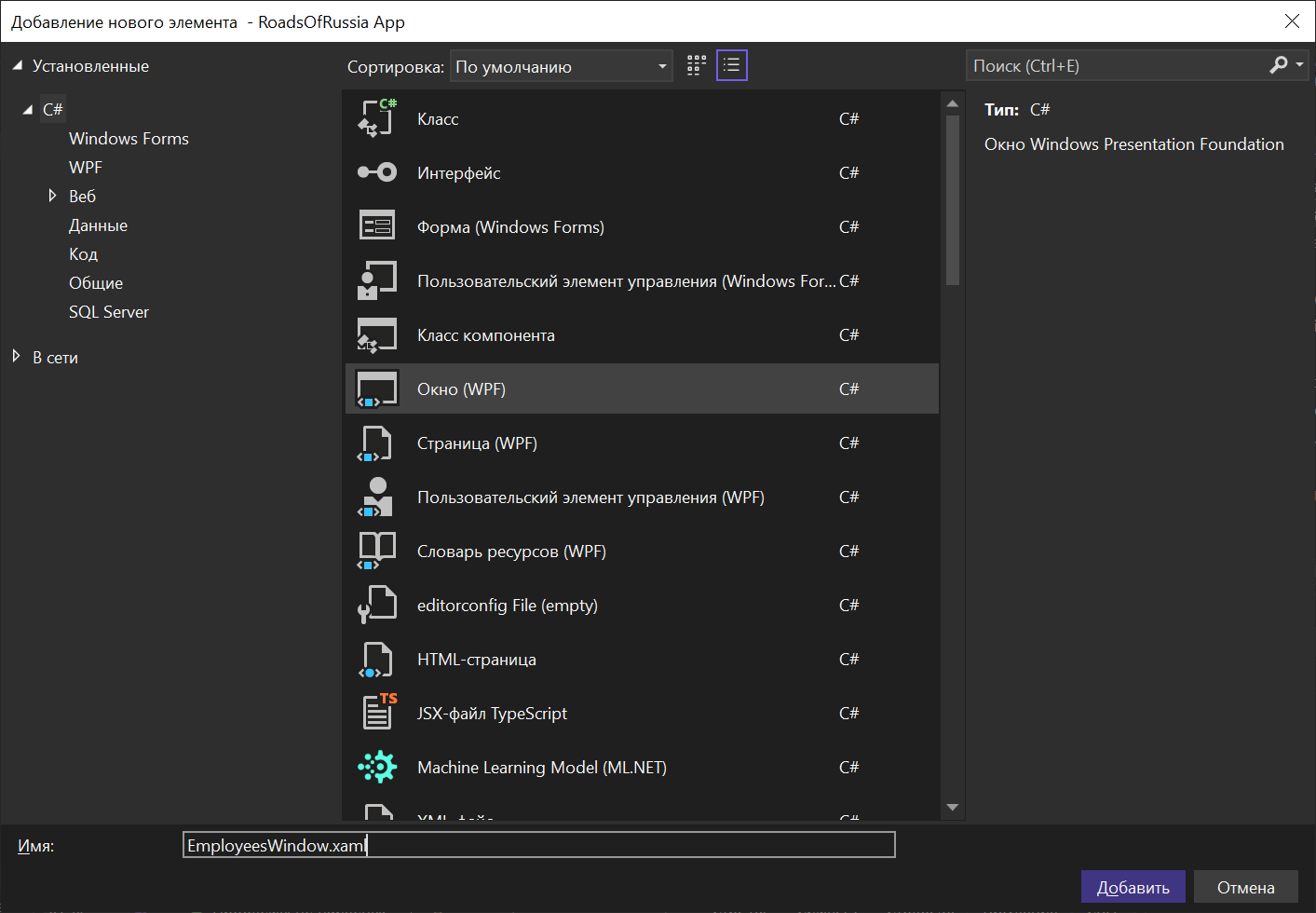


Рисунок 4.4.1 – Создание окна

*Обновление архитектуры:*

В результате добавления нового модуля и расширения существующих модулей архитектура приложения стала более модульной и гибкой.

*Преимущества доработки:*

Повышение стабильности: Модуль обработки ошибок повышает стабильность приложения, предотвращая его аварийное завершение при возникновении ошибок.

Упрощение процесса отладки: Модуль обработки ошибок упрощает процесс отладки приложения.

Доработка архитектуры проекта для интеграции нового модуля позволила повысить стабильность и надежность приложения.

# **Выбраны способы форматирования данных и организована их постобработка,** **транспортные протоколы и форматы сообщений обновлены**

*Форматирование данных:*

В качестве формата хранения данных для сотрудников была использована реляционная база данных SQL Server. Данные хранятся в таблицах с определенными столбцами и типами данных, соответствующими характеристикам сотрудников (имя, фамилия, должность, отдел, филиал, контактная информация и т.д.).

* Структура данных:

Структура данных определена в схеме базы данных (Рисунок 4.1.2). В каждой таблице указаны названия столбцов, типы данных и связи между таблицами.

* Типы данных:

В базе данных используются стандартные типы данных, соответствующие характеру хранимой информации («int» для идентификатора, «nvarchar» для имени и фамилии, «datetime» для даты рождения).

*Постобработка данных:*

* Валидация данных:

Валидация данных осуществляется на уровне приложения.

При вводе данных в форму или при обновлении данных в базе проводится проверка на соответствие установленным правилам (при вводе телефонного номера проверяется его длина).

* Преобразование данных:

Преобразование данных происходит в некоторых случаях при передаче данных между компонентами приложения или при выводе данных на экран (при преобразовании даты из формата дата в базе данных в формат строки для вывода на экран).

*Транспортные протоколы и форматы сообщений:*

* Протокол передачи данных:

В приложении используется прямое соединение с базой данных без использования API или сетевых протоколов.

Данные передаются непосредственно из приложения в базу данных и обратно.

* Формат сообщений:

Формат сообщений определяется структурой данных в базе данных.

Данные передаются в виде строковых значений и затем преобразуются в соответствующие типы данных на стороне приложения.

*Обновление транспортных протоколов и форматов сообщений:*

* Новые форматы сообщений:

В приложении не используются специальные форматы сообщений для обмена данными между компонентами.

Данные передаются в виде строковых значений и преобразуются в соответствующие типы данных на стороне приложения.

В итоге, изменения в обработке данных и обновление запросов к базе данных позволили обеспечить корректную интеграцию нового модуля в существующее приложение, гарантируя стабильную работу приложения и корректное отображение информации о сотрудниках.

# **Выполнена доработка модуля и дополнительная обработка исключительных ситуаций, в том числе, с созданием классов-исключений**

Для добавления новых функций, было создано новое окно (EmployessWindow), на котором будет выводится дополнительная информация о выбранном сотруднике и формироваться персональный QR-код данного сотрудника. Разработка данного окна представлена на рисунке 4.6.1.

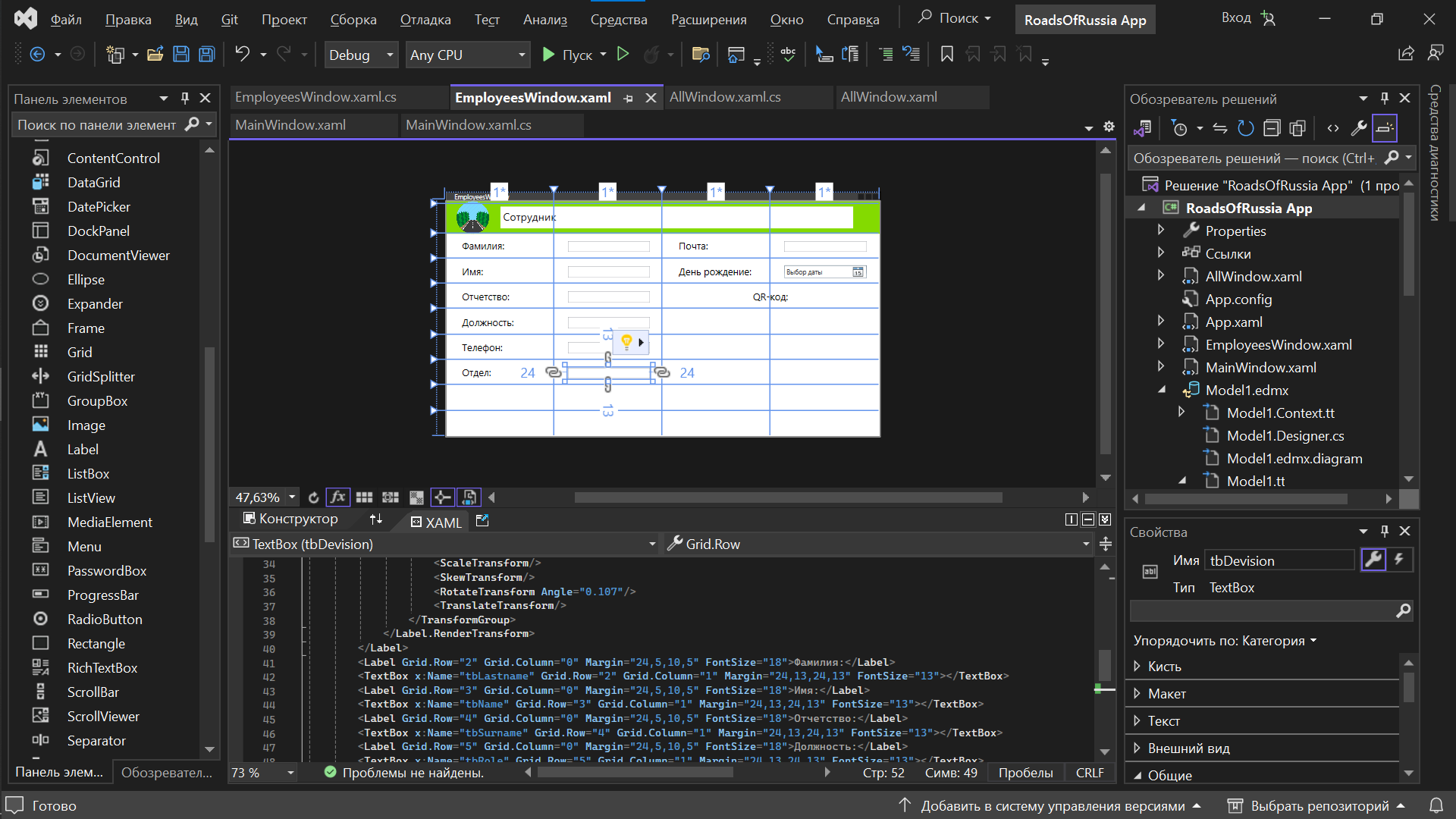


Рисунок 4.6.1 – Разработка окна

На окне выводится следующая информация: ФИО, должность, телефон, отдел, почта, дата рождения и QR-код (Рисунок 4.6.2).

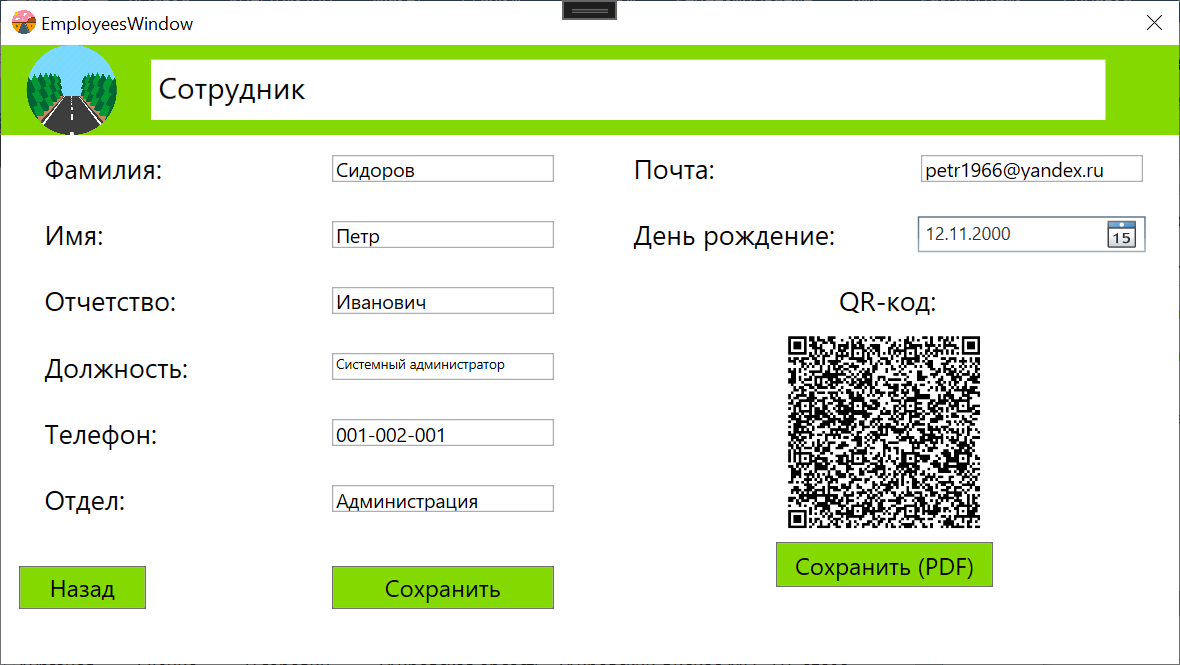


Рисунок 4.6.2 – Конечный результат

QR-код можно сохранить в документ формата PDF (Рисунок 4.6.3).

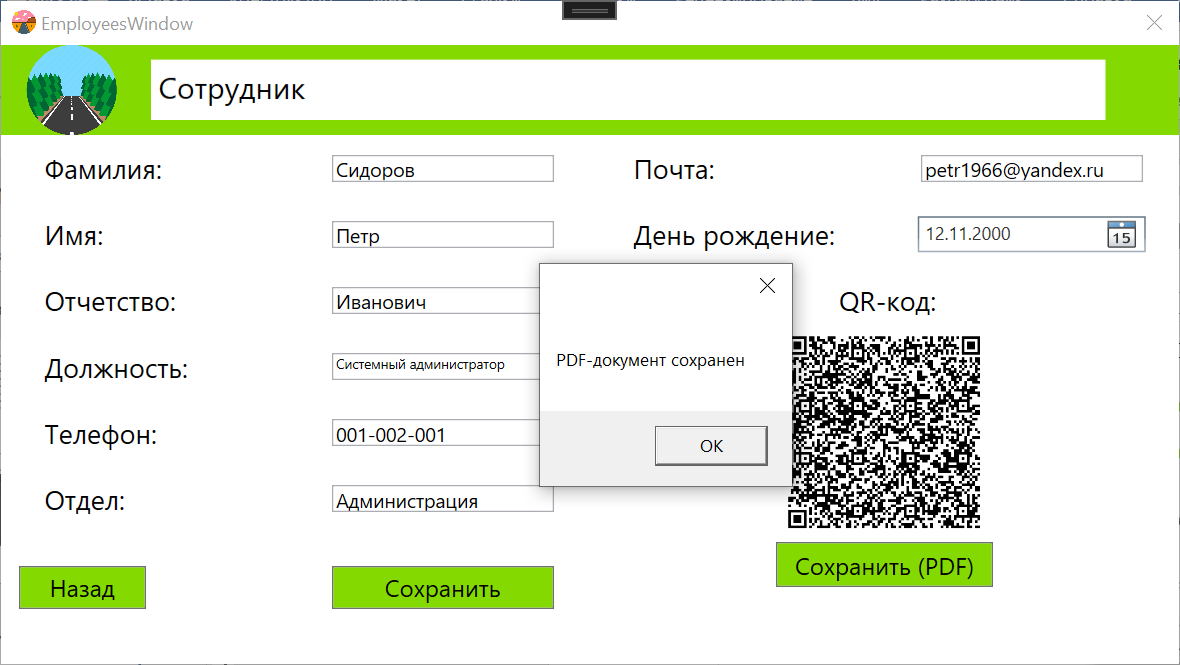


Рисунок 4.6.3 – Сохранение QR-код в PDF документ

PDF документ с QR-кодом сохраняется в папку с проектом (Рисунок 4.6.4). Название документа состоит из pdf, Doc и почты сотрудника – petr1966@yandex.ru.



Рисунок 4.6.4 – Результат сохранения QR-кода

Если по каким-то причинам нет подключения к базе данных (она не доступна, удалена или переименована), чтобы приложение критически не завершала свою работу, в коде прописана конструкция try{} catch{}. С обязательным пояснение, почему не удается подключиться к базе данных. Код конструкции try{} catch{} продемонстрирован на рисунке 4.6.5.

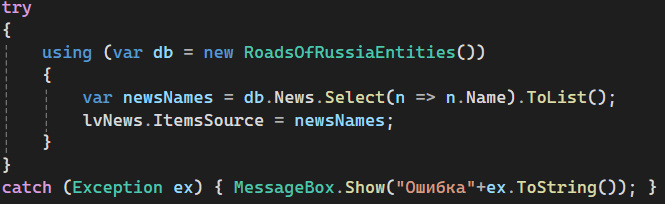


Рисунок 4.6.5 – Конструкция try{} catch{}

В приложении были использованы не только конструкции try{} catch{} по и конструкция if () else (Рисунок 4.6.6).

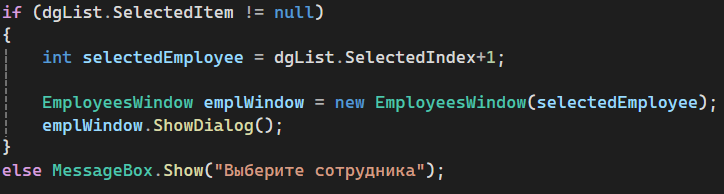


Рисунок 4.6.6 – конструкция if () else

В данном примере if () else используется для того, чтобы определить выбрал пользователь в списке сотрудников кого-то или нет. Если выбрал, то данные сотрудника передаются на следующее окно, которое отображает дополнительную информацию о сотруднике и содержит персональный QR-код с возможностью его дальнейшего сохранения в PDF формат.

# **Определены качественные показатели полученного проекта**

Качественные показатели для разработанного проекта в Visual Studio:

* Безопасность.

Защита от атак по перебору паролей - ограничение количества попыток ввода (3 попытки).

* Удобство.

Интуитивно понятный интерфейс для входа в систему.

* Корректность данных.

Отображение актуальных данных о сотрудниках.

Отсутствие ошибок и дубликатов в данных.

* Эффективность.

Быстрая загрузка списка сотрудников.

Возможность фильтрации по нескольким критериям (область, филиал и отдел).

Возможность поиска по ФИО, должности, отделу и почте.

* Удобство.

Четкое и структурированное отображение информации о сотруднике.

Возможность редактирования данных о сотруднике.

* Эстетика.

Приятный дизайн окна с информацией о сотруднике.

Соответствие общей стилистике приложения.

* Корректность.

Создание QR-кода с правильной информацией о сотруднике.

* Эффективность.

Быстрое формирование QR-кода.

Возможность сохранения QR-кода в формате PDF.

* Производительность.

Быстрая загрузка приложения и выполнение операций (авторизация, загрузка списка сотрудников, генерация QR-кода).

Оптимальное использование памяти и процессорного времени.

* Надежность.

Устойчивость к ошибкам: Отсутствие сбоев и ошибок в работе приложения.

Восстановление: Возможность восстановления после сбоев или ошибок.

* Удобство использования.

Интуитивно понятный интерфейс: Простой и понятный интерфейс для всех пользователей.

Доступность: Приложение доступно для пользователей с различными уровнями технических знаний.

* Безопасность.

Защита данных: Защита данных о сотрудниках от несанкционированного доступа.

Конфиденциальность: Обеспечение конфиденциальности данных о сотрудниках.

# **Результат интеграции сохранен в системе контроля версий**

Результат интеграции нового модуля был зафиксирован в системе контроля версий Git с использованием GitHub Desktop (Рисунок 4.8.1).

Был создан коммит, включающий все изменения, связанные с интеграцией нового модуля. В комментарии к коммиту была предоставлена подробная информация о внесенных изменениях.

В репозитории GitHub теперь доступна история изменений, которая отражает все этапы интеграции нового модуля, включая создание новых файлов, внесение изменений в существующий код и добавление комментариев.

Использование системы контроля версий Git позволило зафиксировать результат интеграции нового модуля и отслеживать историю изменений. Это гарантирует прозрачность и безопасность процесса разработки и позволяет легко управлять изменениями в коде проекта.

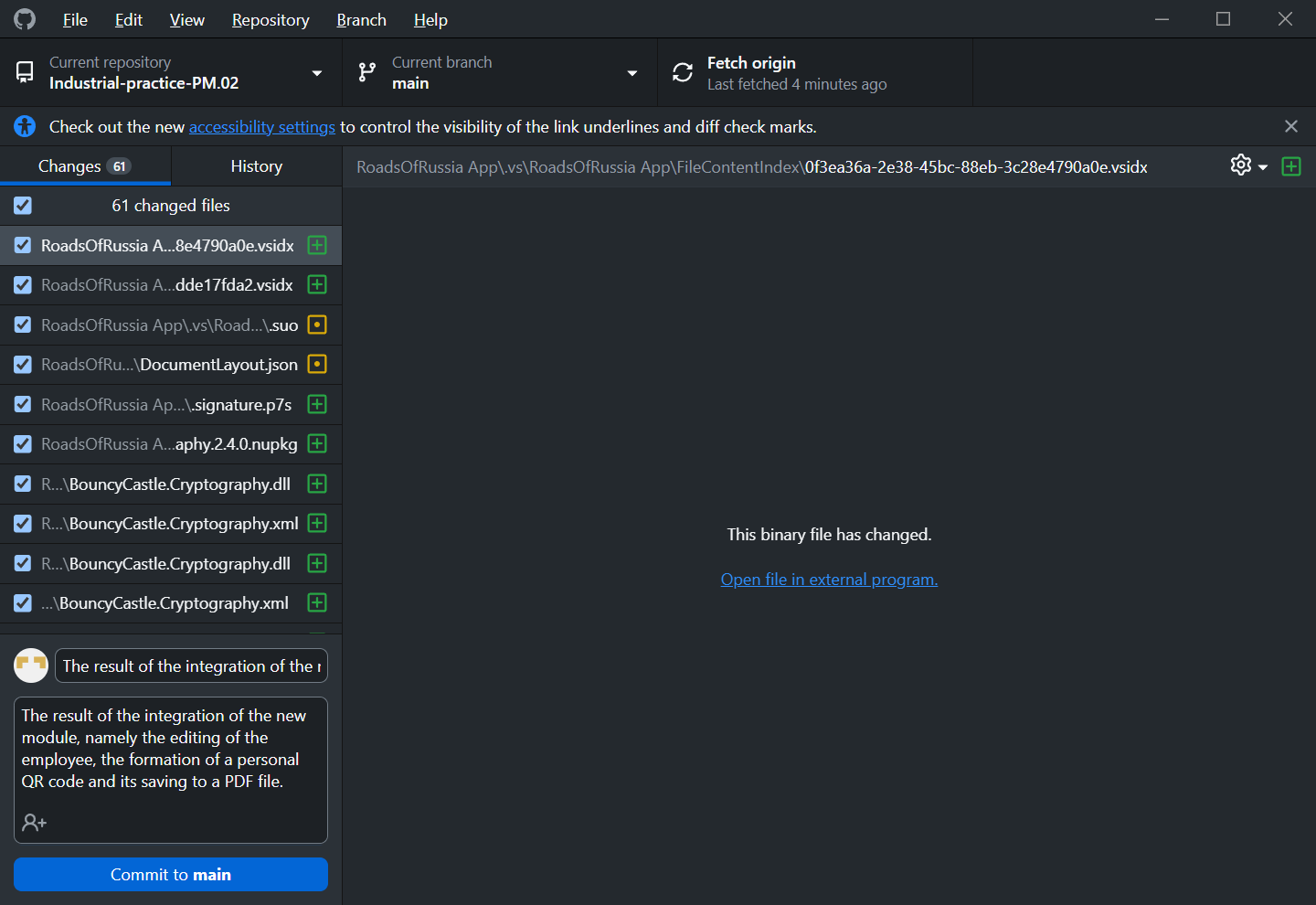


Рисунок 4.8.1 – Сохранение результатов

# **Протестирована интеграция модулей проекта и выполнена отладка проекта с применением инструментальных средств среды**

Для отладки в Visual Studio, был выбран режим Debug для того, чтобы отслеживать ошибки в написанном коде.

Проверка выводимой информации о сотрудник (Рисунок 4.9.1).

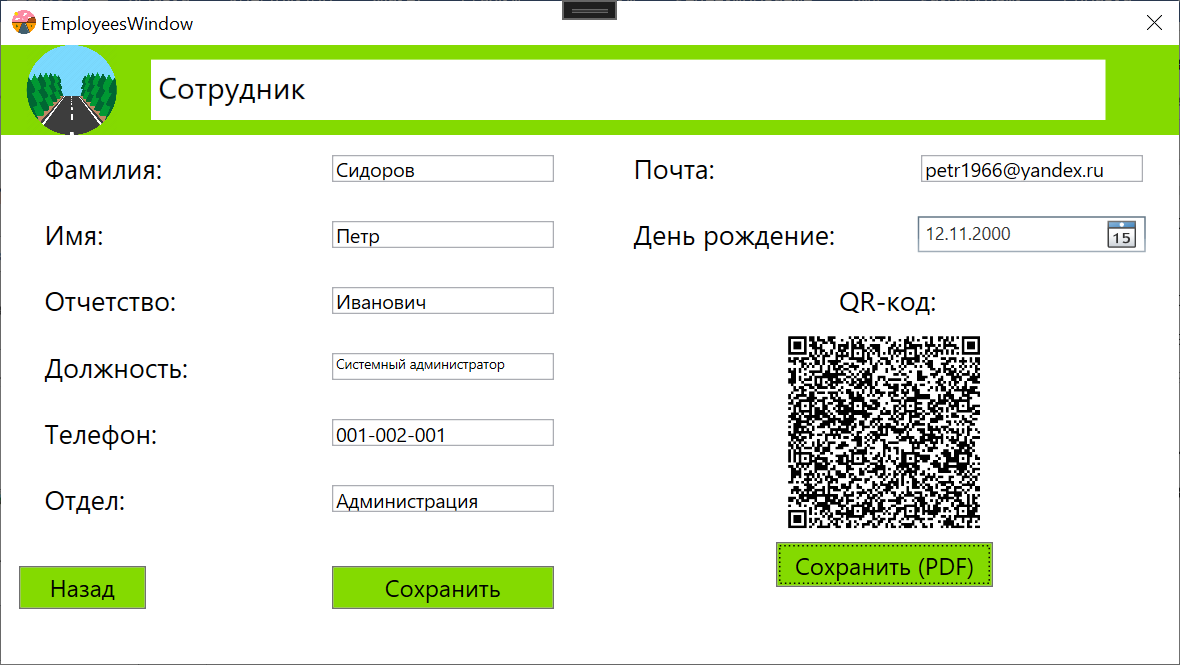


Рисунок 4.9.1 – Проверка информации

Проверка сохранения измененной информации о сотруднике в базу данных (Рисунок 4.9.2).

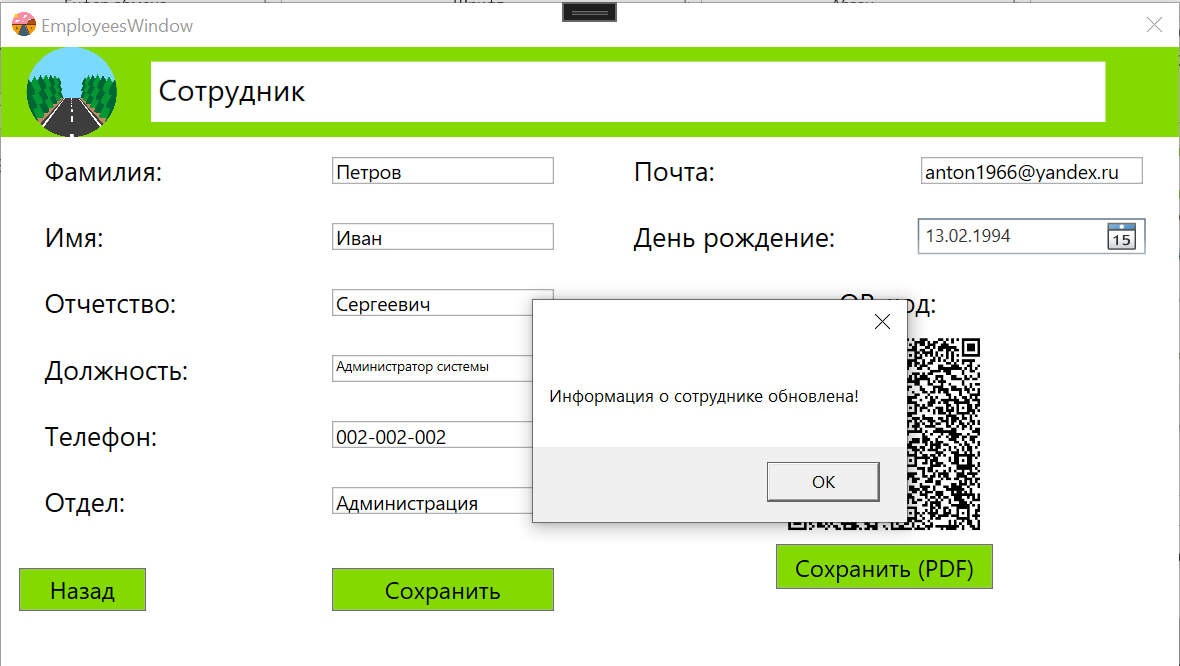
****

Рисунок 4.9.2 – Проверка сохранения в базу данных

Проверка правильного отображения информации о сотруднике в QR-коде (Рисунок 4.9.3).



Рисунок 4.9.3 – Проверка QR-кода

Проверка сохранения QR-код в папку проекта (Рисунок 4.9.4).

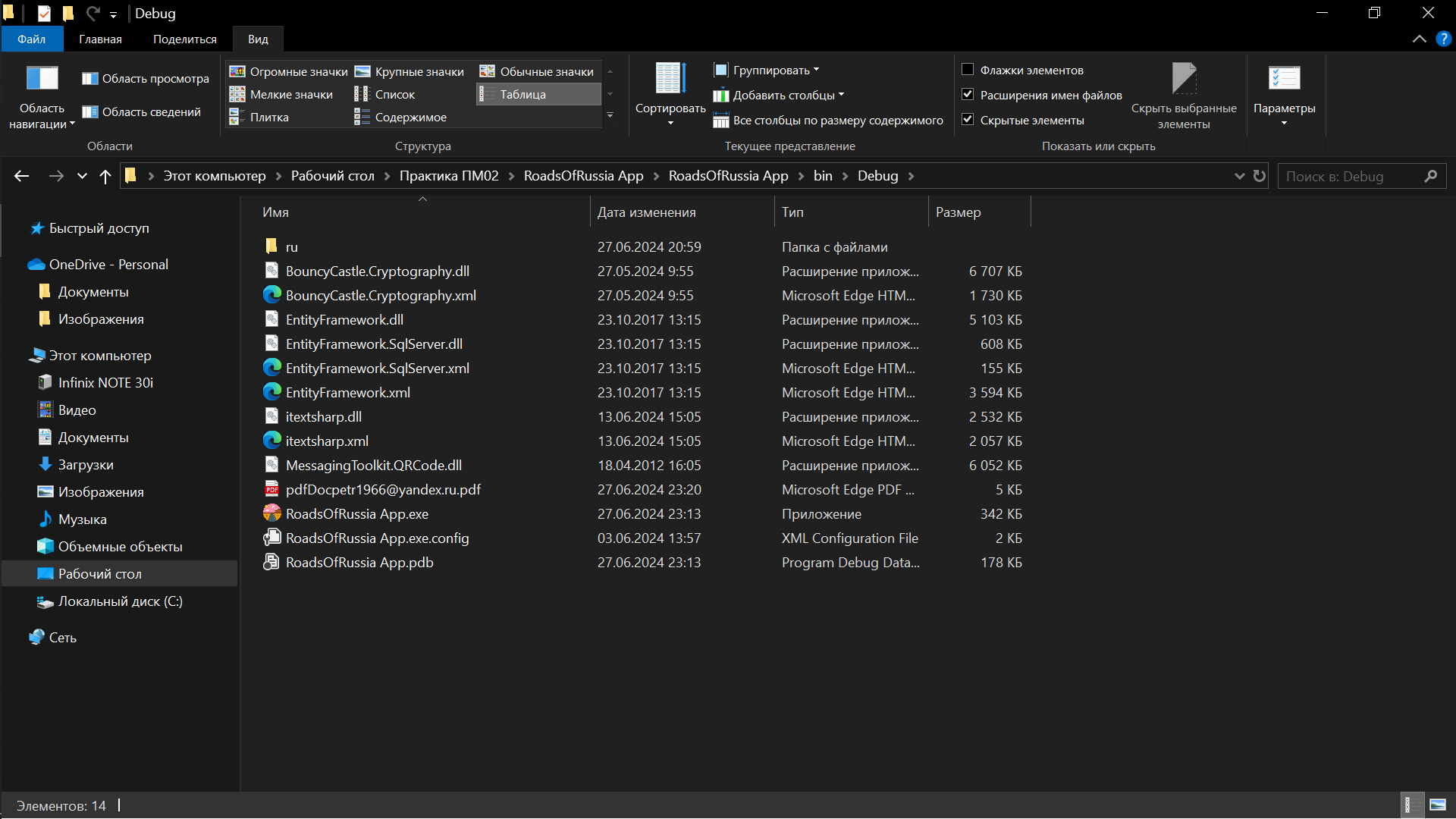


Рисунок 4.9.4 – Проверка сохранения QR-кода

# **Разработан тестовый сценарий и тестовые пакеты**

Разработка TestCase.

TestCase №1

Таблица 1 – TestCase №1

|  |  |
| --- | --- |
| **TestCase №** | 1 |
| **Приоритет теста** | Высокий |
| **Название тестирования/Имя** | Авторизация с корректными данными |
| **Резюме испытания** | Проверка успешной авторизации пользователя |
| **Шаги тестирования** | Ввести корректный логин и пароль в поля авторизации.  Нажать кнопку «Войти» |
| **Данные тестирования** | Логин: 1  Пароль: 11 |
| **Ожидаемый результат** | Пользователь успешно авторизован и переходит в главное окно приложения |
| **Фактический результат** | Пользователь успешно авторизован и переходит в главное окно приложения |
| **Предпосылки** | Приложение запущено и доступно для авторизации |
| **Постусловия** | Пользователь авторизован в системе |
| **Статус (Pass/Fail)** | Pass |
| **Комментарии** | Тест успешно выполнен |

TestCase №2

Таблица 2 – TestCase №2

|  |  |
| --- | --- |
| **TestCase №** | 2 |
| **Приоритет теста** | Высокий |
| **Название тестирования/Имя** | Проверка поиска по полному совпадению фамилии |
| **Резюме испытания** | Проверка, что система корректно выводит список сотрудников с заданной фамилией при полном совпадении |
| **Шаги тестирования** | Авторизация.  Ввод фамилии «Иванов» в строку поиска |
| **Данные тестирования** | Иванов |
| **Ожидаемый результат** | Список сотрудников с фамилией «Иванов» отображается корректно |
| **Фактический результат** | Список сотрудников с фамилией «Иванов» отображается корректно |
| **Предпосылки** | При загрузке приложения должен выгружаться список всех сотрудников |
| **Постусловия** |  |
| **Статус (Pass/Fail)** | Pass |
| **Комментарии** | Тест успешно выполнен |

TestCase №3

Таблица 3 – TestCase №3

|  |  |
| --- | --- |
| **TestCase №** | 3 |
| **Приоритет теста** | Средний |
| **Название тестирования/Имя** | Проверка поиска по частичному совпадению фамилии |
| **Резюме испытания** | Проверка, что система корректно выводит список сотрудников, у которых в фамилии встречается заданный набор букв |
| **Шаги тестирования** | Авторизация.  Ввод фамилии «Ива» в строку поиска |
| **Данные тестирования** | Ива |
| **Ожидаемый результат** | Список сотрудников с частичным совпадением фамилии «Ива» отображается корректно |
| **Фактический результат** | Список сотрудников с частичным совпадением фамилии «Ива» отображается корректно |
| **Предпосылки** | При загрузке приложения должен выгружаться список всех сотрудников |
| **Постусловия** |  |
| **Статус (Pass/Fail)** | Pass |
| **Комментарии** | Тест успешно выполнен |

TestCase №4

Таблица 4 – TestCase №4

|  |  |
| --- | --- |
| **TestCase №** | 4 |
| **Приоритет теста** | Средний |
| **Название тестирования/Имя** | Проверка поиска по несуществующей фамилии |
| **Резюме испытания** | Проверка, что система выводит пустой список |
| **Шаги тестирования** | Авторизация.  Ввод фамилии «Jon» в строку поиска |
| **Данные тестирования** | Jon |
| **Ожидаемый результат** | Отображается пустой список |
| **Фактический результат** | Отображается пустой список |
| **Предпосылки** | При загрузке приложения должен выгружаться список всех сотрудников |
| **Постусловия** |  |
| **Статус (Pass/Fail)** | Pass |
| **Комментарии** | Тест успешно выполнен |

TestCase №5

Таблица 5 – TestCase №5

|  |  |
| --- | --- |
| **TestCase №** | 5 |
| **Приоритет теста** | Высокий |
| **Название тестирования/Имя** | Просмотр информации о сотруднике |
| **Резюме испытания** | Проверка открытия нового окна с информацией о выбранном сотруднике |
| **Шаги тестирования** | Авторизоваться в системе.  Выбрать запись с сотрудником в таблице |
| **Данные тестирования** | Первый сотрудник из списка |
| **Ожидаемый результат** | Открывается новое окно с подробной информацией о выбранном сотруднике |
| **Фактический результат** | Открывается новое окно с подробной информацией о выбранном сотруднике |
| **Предпосылки** | Пользователь авторизован в системе, список всех сотрудников загружен |
| **Постусловия** | Пользователь находится в новом окне с информацией о сотруднике |
| **Статус (Pass/Fail)** | Pass |
| **Комментарии** | Тест успешно выполнен |

# **Выполнено тестирование интеграции и ручное тестирование**

**Функциональное тестирование.**

*Авторизация.*

Присутствует ограничение попыток входа. Результат первого неверно введенного пароля представлен на рисунке 4.11.1.

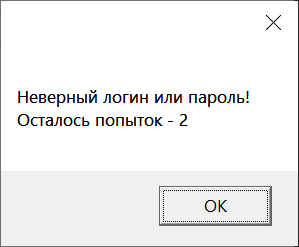


Рисунок 4.11.1 – Результат тестирования

Результат второго неверно введенного пароля представлен на рисунке 4.11.2.

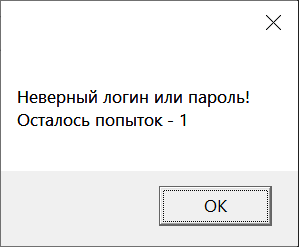


Рисунок 4.11.2 – Результат тестирования

Результат третьего неверно введенного пароля представлен на рисунке 4.11.3.

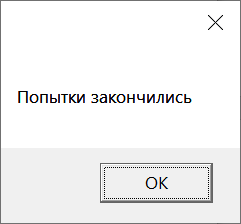


Рисунок 4.11.3 – Результат тестирования

После трех неверно введенных логина или пароля на окне авторизации исчезает кнопка «Войти» (Рисунок 4.11.4).

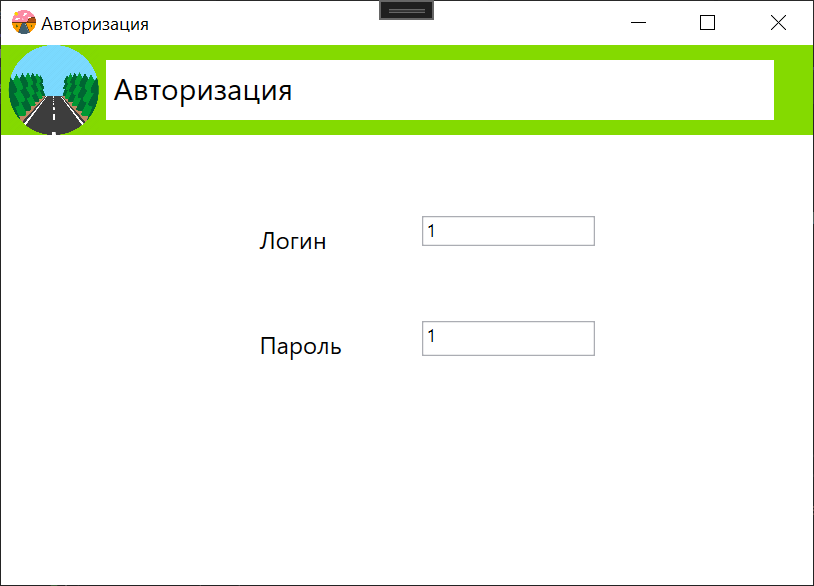


Рисунок 4.11.4 – результат тестирования

*Фильтрация списка сотрудников.*

Выбор всех сотрудников из Кировской области представлен на рисунке 4.11.5.

Выбор всех сотрудников из Кировской области и Кировского филиала №1 представлен на рисунке 4.11.6.

Выбор всех сотрудников из Кировской области и Кировского филиала №2 представлен на рисунке 4.11.7.

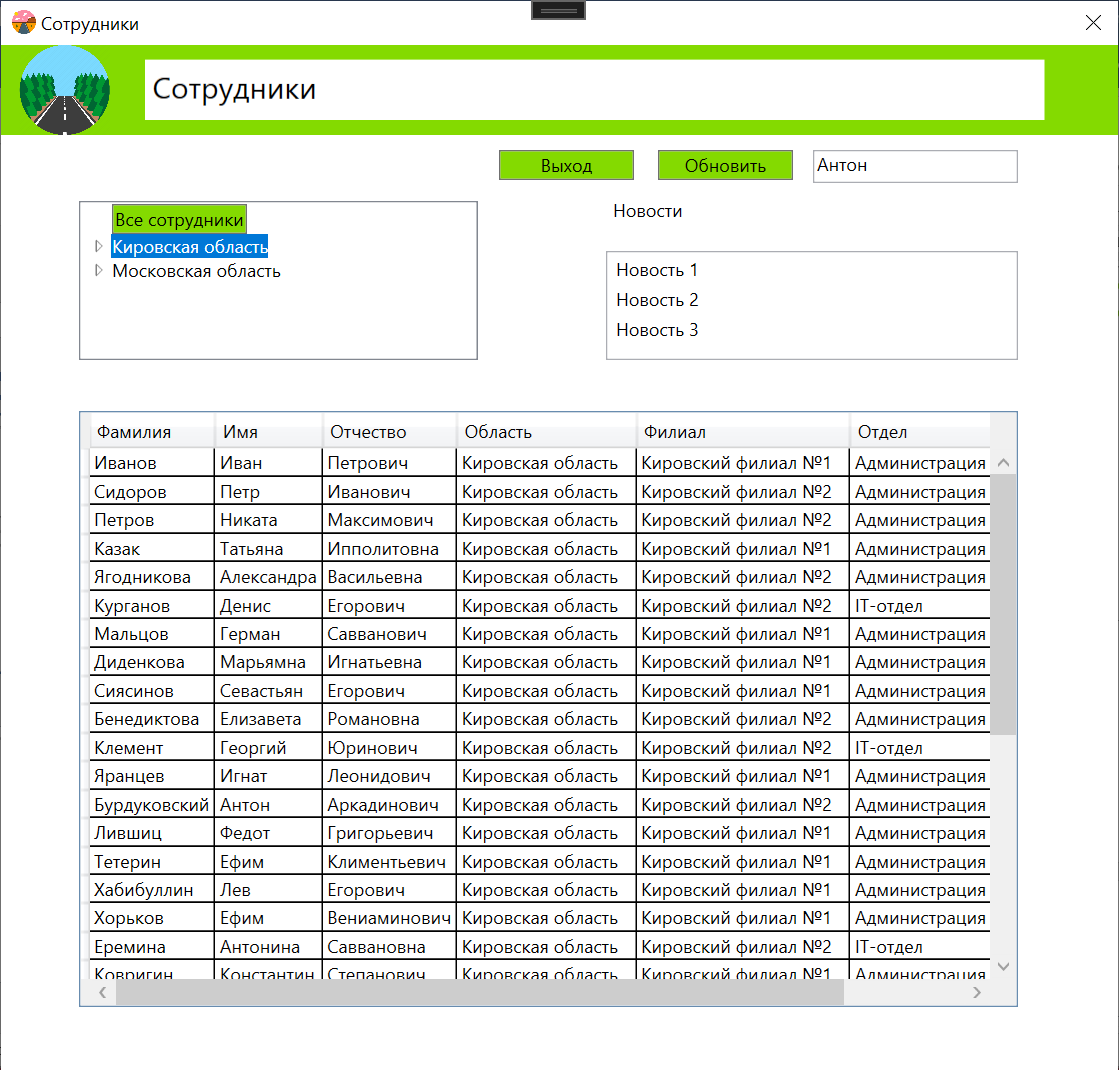


Рисунок 4.11.5 – Результат тестирования



Рисунок 4.11.6 – Результат тестирования

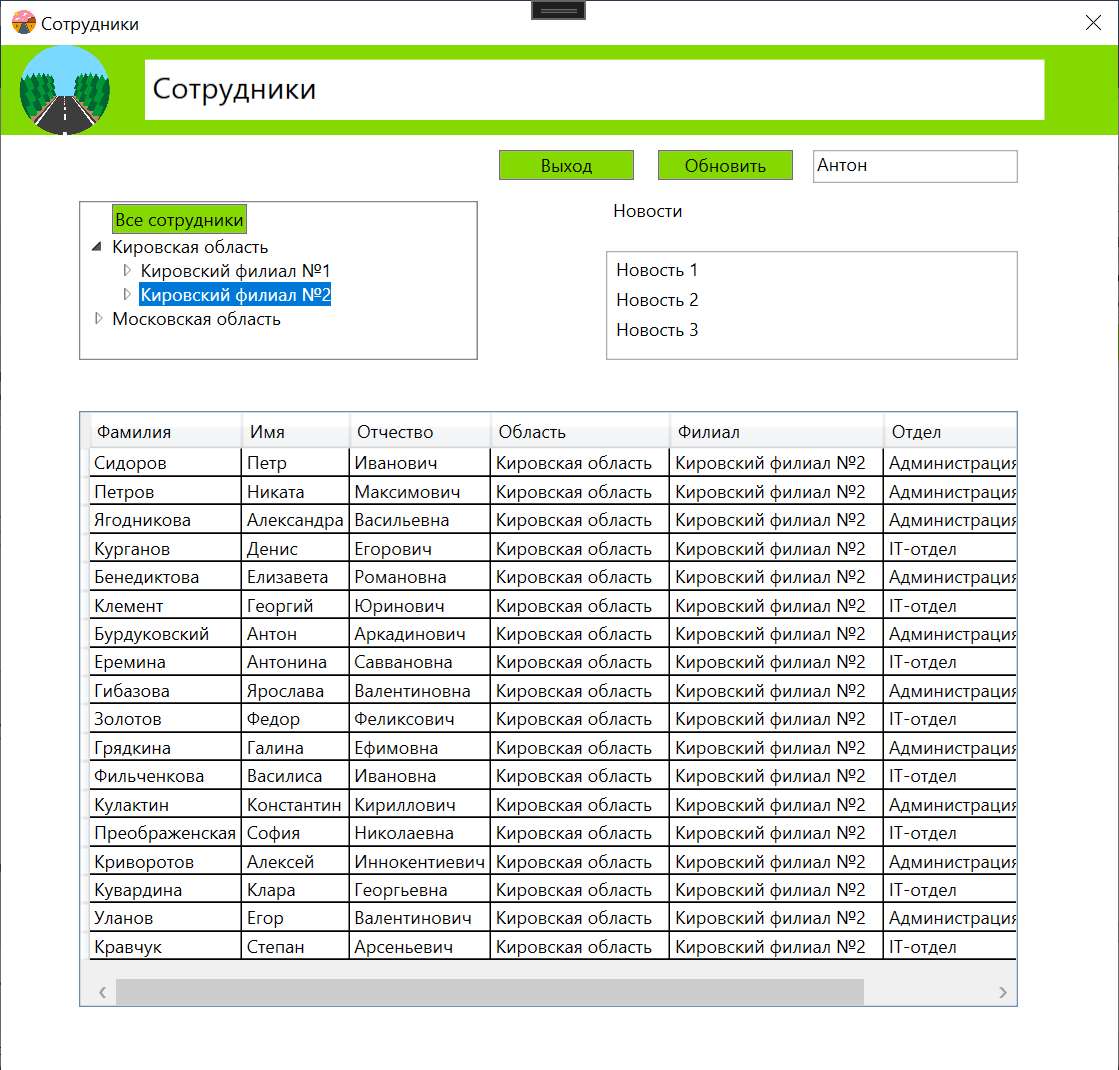


Рисунок 4.11.7 – Результат тестирования

Выбор всех сотрудников из администрации Кировского филиала №2 Кировской области представлен на рисунке 4.11.8.

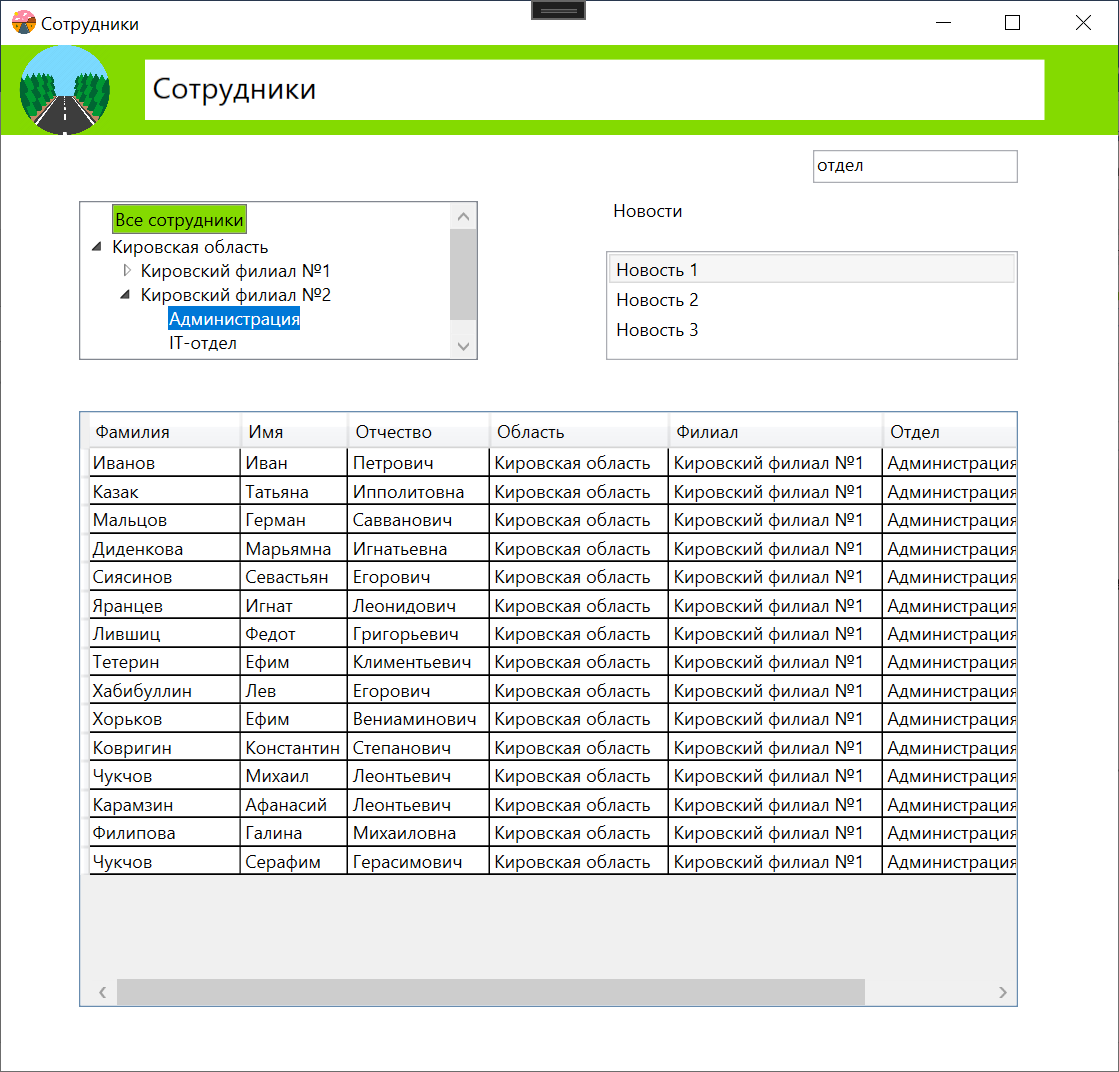


Рисунок 4.11.8 – Результат тестирования

Выбор всех сотрудников из Московской области представлен на рисунке 4.11.9.

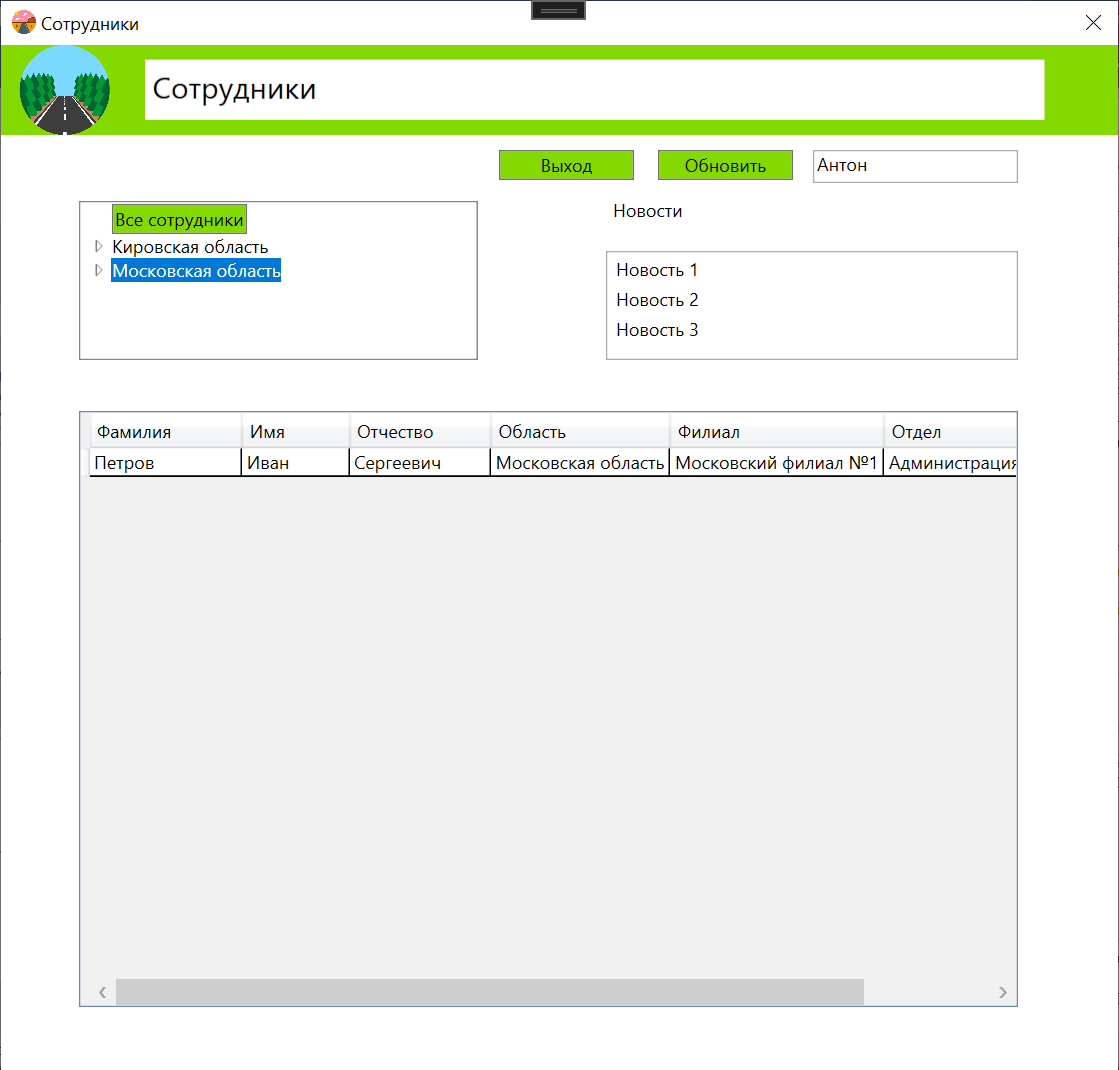


Рисунок 4.11.9 – Результат тестирования

Если при выборе нажать на Кировский филиал №1, но при этом не выбрать область, то список сотрудников будет пуст (Рисунок 4.11.10).

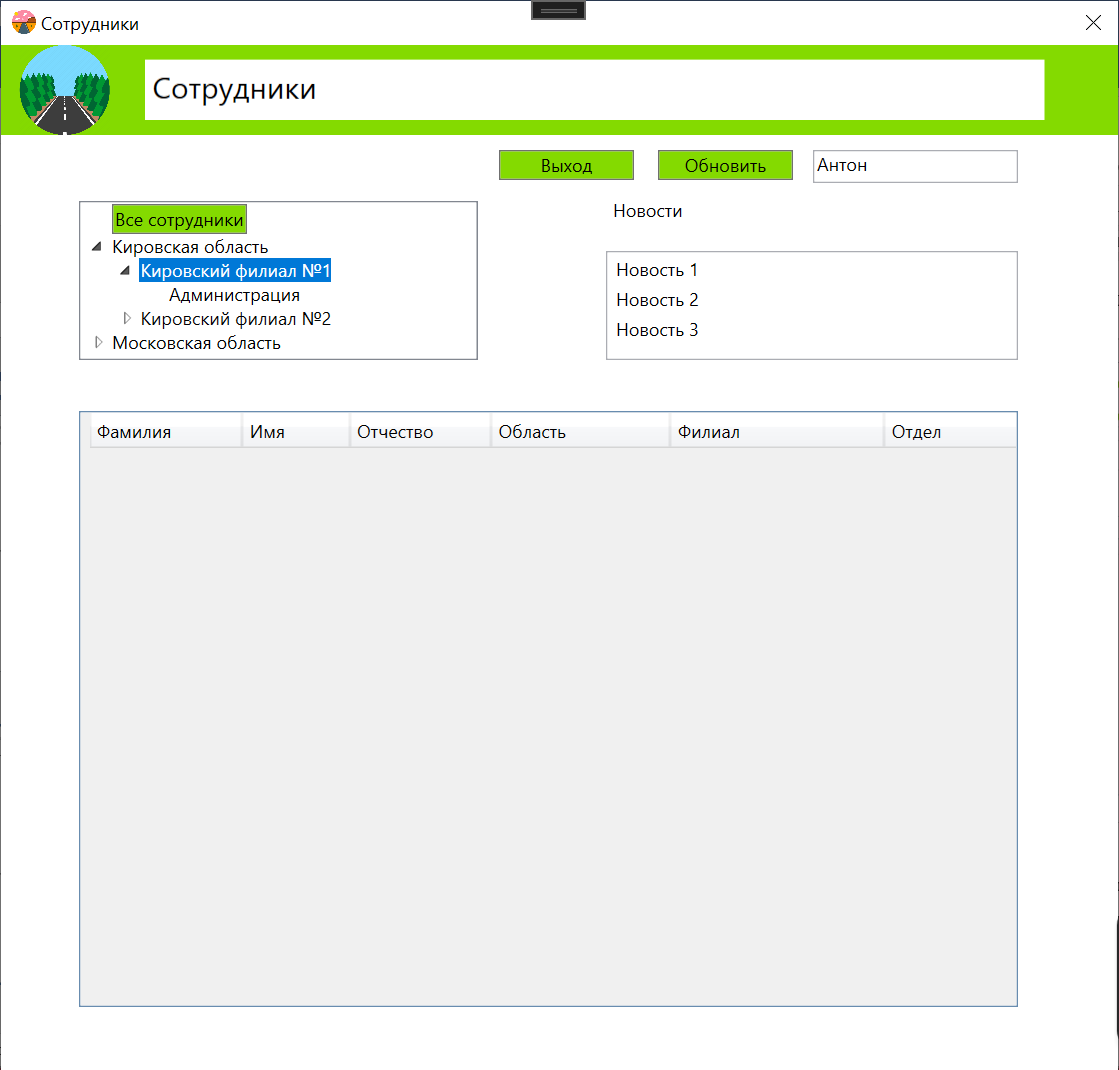


Рисунок 4.11.10 – Результат тестирования

Результат тестирования кнопки «Все сотрудники» представлен на рисунке 4.11.11.

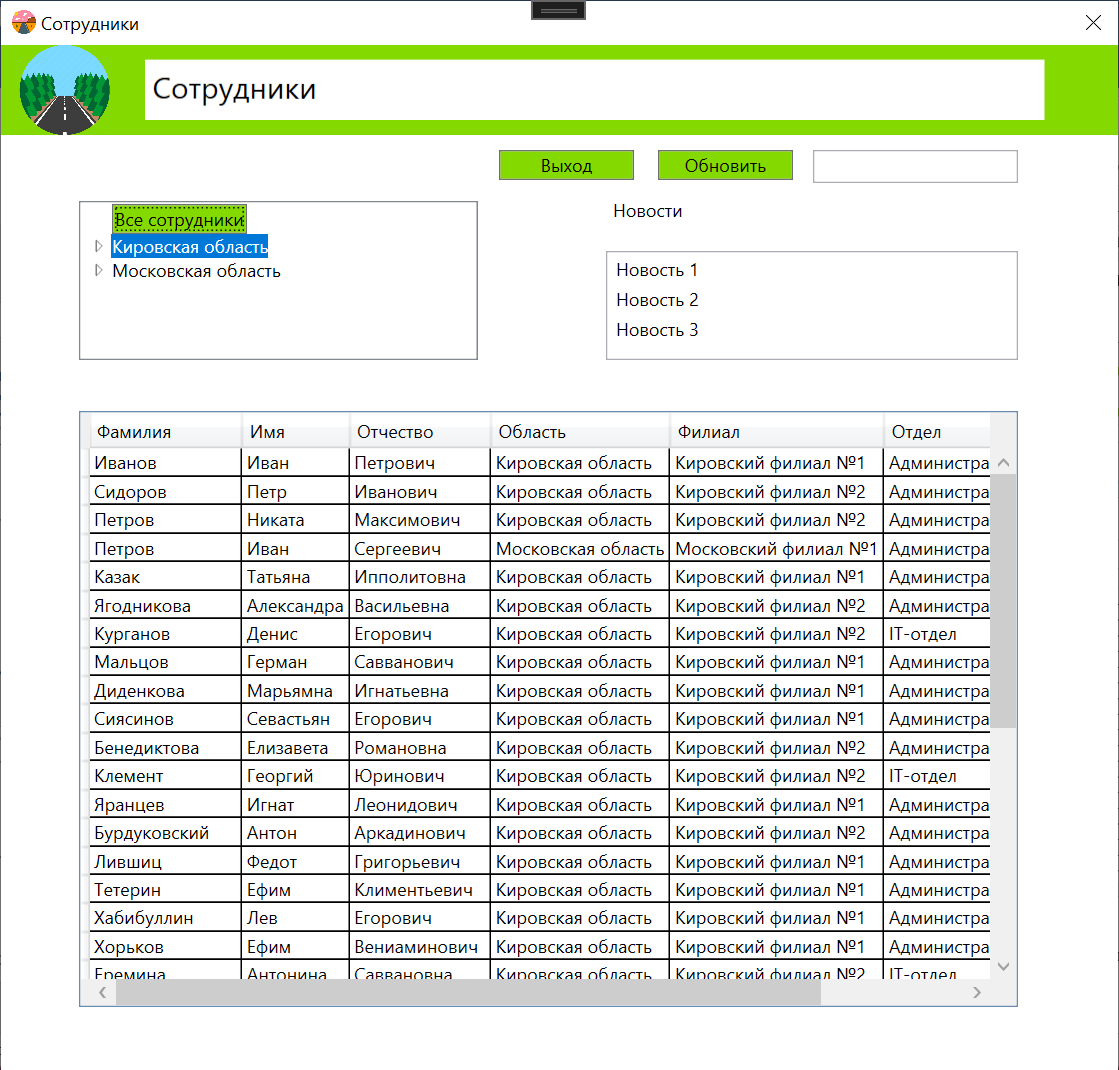


Рисунок 4.11.11 – Результат тестирования

*Функция поиска.*

Поиск работает сразу при вводе символов в строку поиска. Результат поиска по имени (Иван) представлен на рисунке 4.11.12.

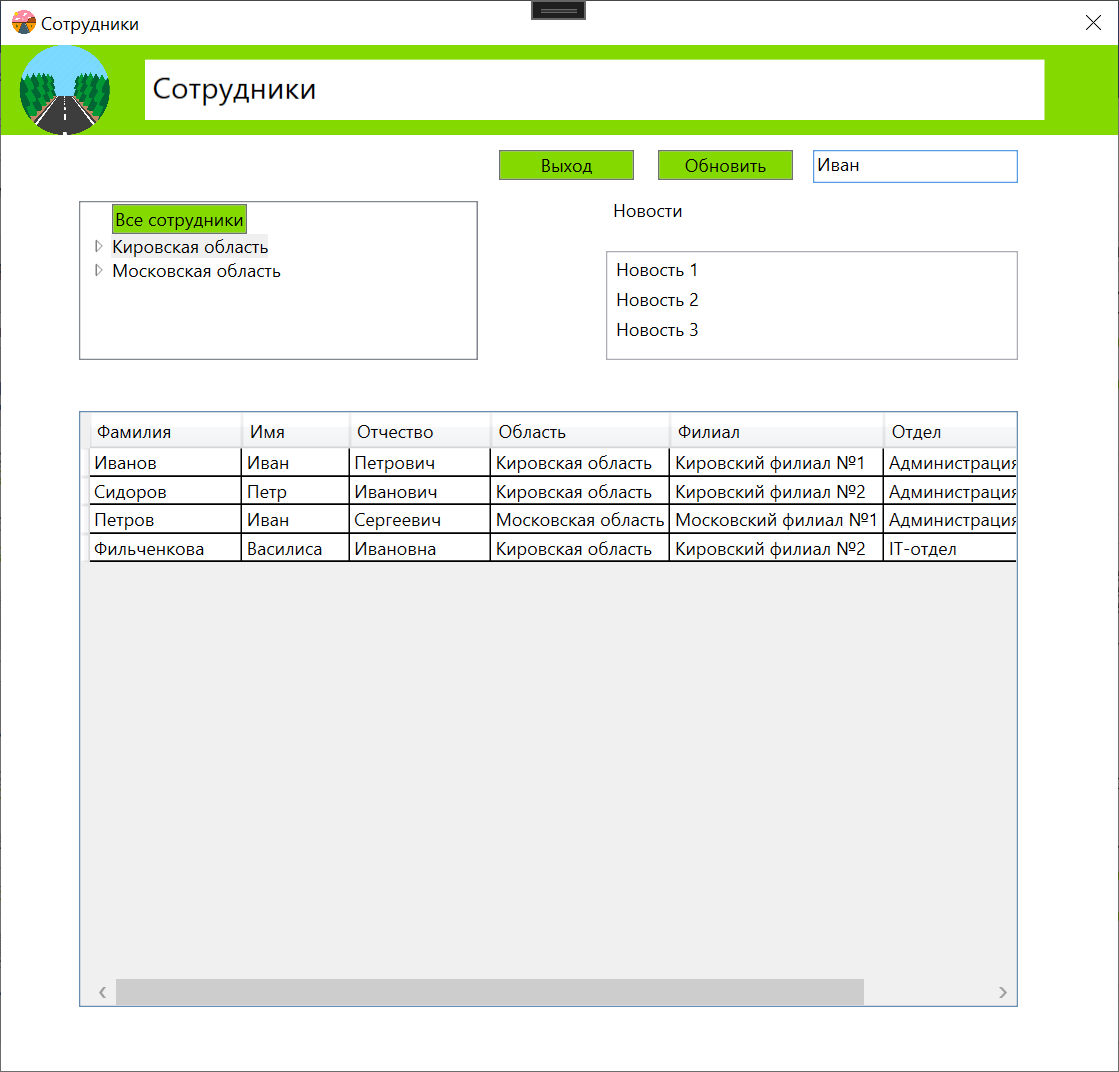


Рисунок 4.11.12 – Результат тестирования

Поиск осуществляется не только по ФИО, но и по названию отделов, в которых работают сотрудники (Рисунок 4.11.13).

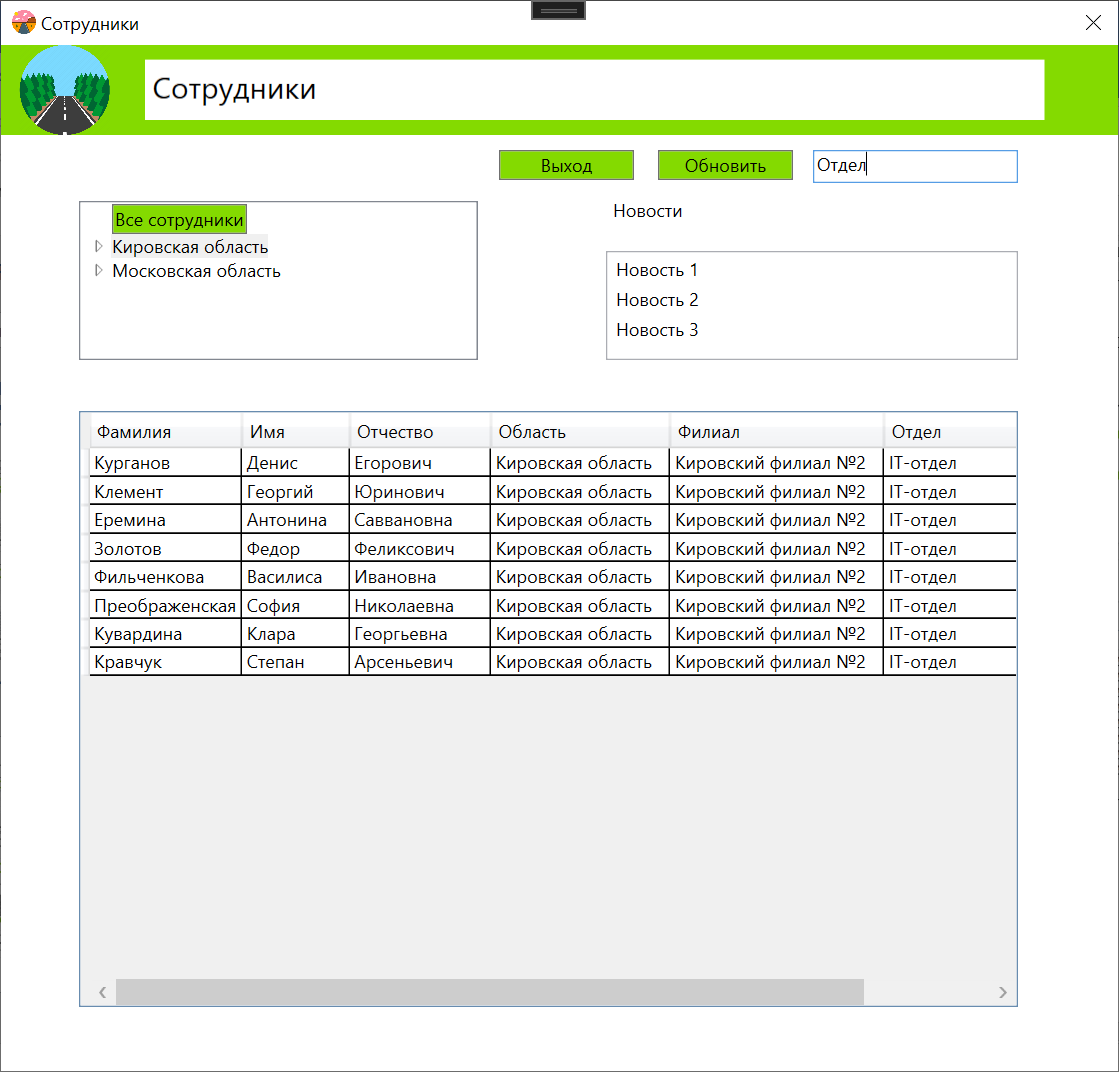


Рисунок 4.11.13 – Результат тестирования

Если мы попытаемся ввести в строку поиска фамилию на английском языке, то список сотрудников будет пуст, потому что в базе данных сотрудников с фамилией, написанной на английском языке нет. Результат данного тестирования изображен на рисунке 4.11.14.

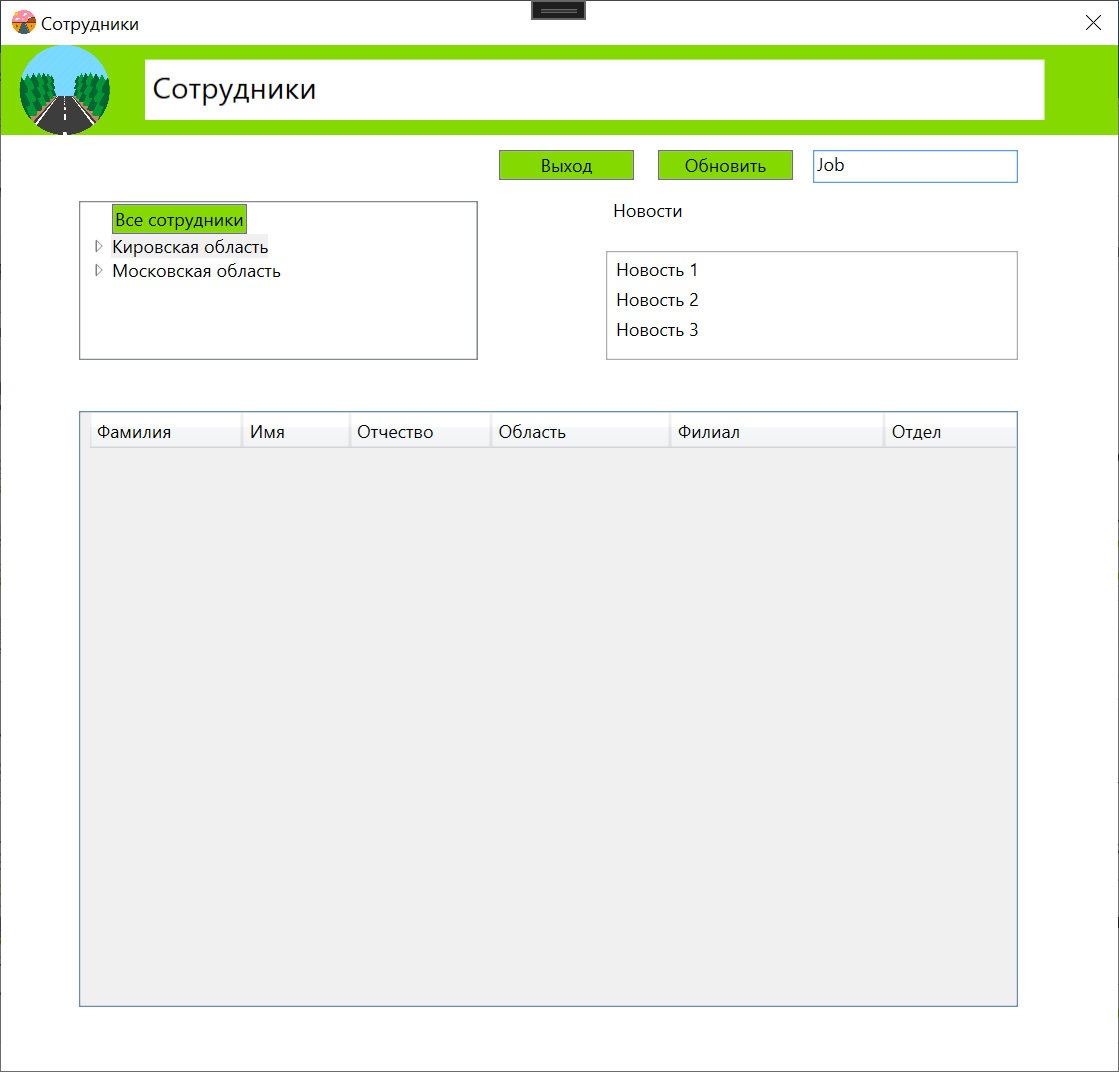


Рисунок 4.11.14 – Результаты тестирования

*Отображение основной информации о сотруднике.*

Выбираем сотрудника, нажимаем на его запись. Вся информация выводится корректно (Рисунок 4.11.15).

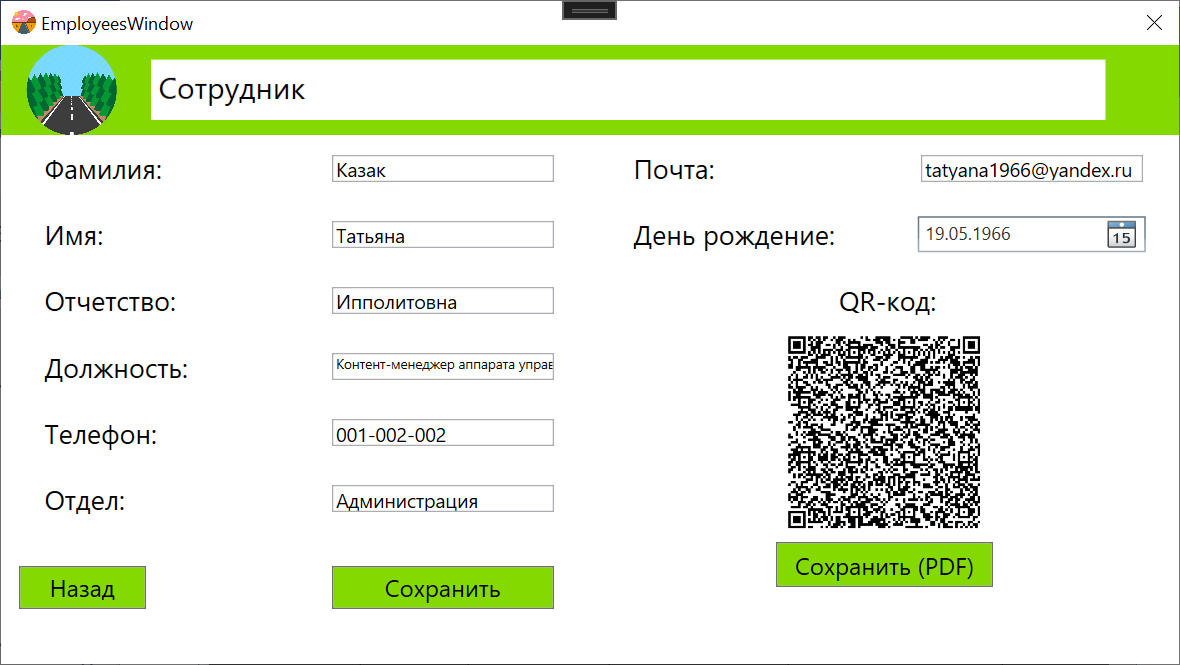


Рисунок 4.11.15 – Результат тестирования

*Сохранение измененной информации о сотруднике.*

Попытаемся изменить имя сотрудника с Татьяны на Ольгу. При нажатии на кнопку «Сохранить» выходит сообщение о том, что информация обновлена (Рисунок 4.11.16).

На рисунке 4.11.17 показана обновленная информация о сотруднике в базе данных.

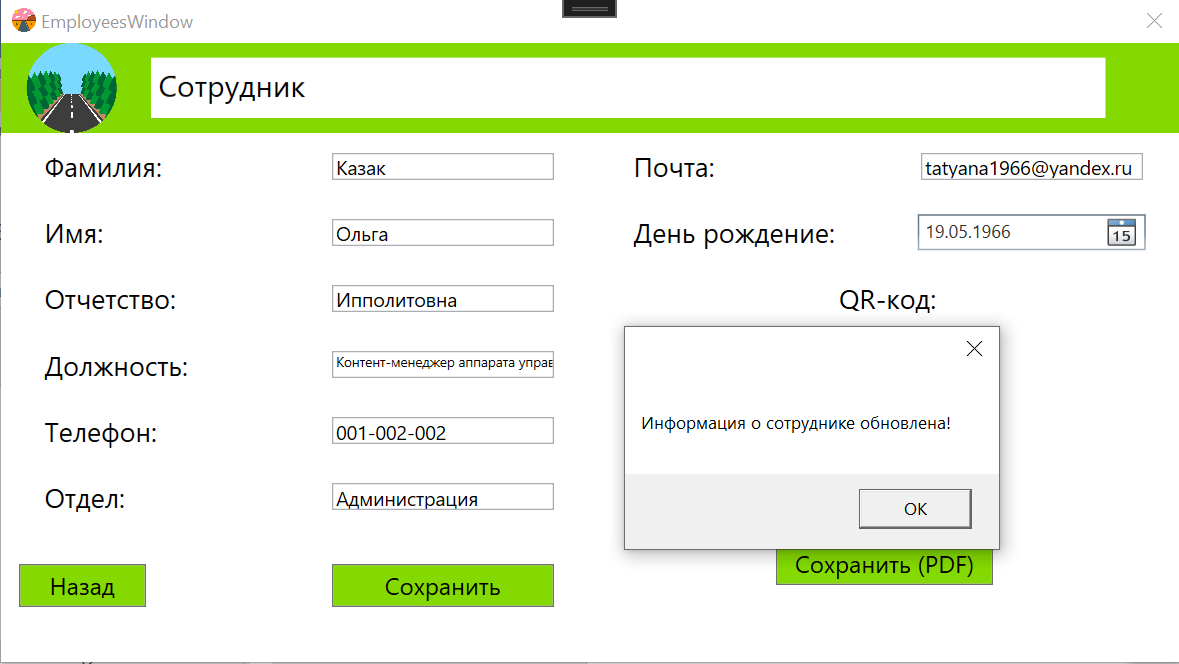


Рисунок 4.11.16 – Результат тестирования



Рисунок 4.11.17 – Результат тестирования

*Сохранение QR-кода.*

При нажатии на кнопку «Сохранить (PDF)» под QR-кодом выходит сообщение о том, что документ сохранен (Рисунок 4.11.18).

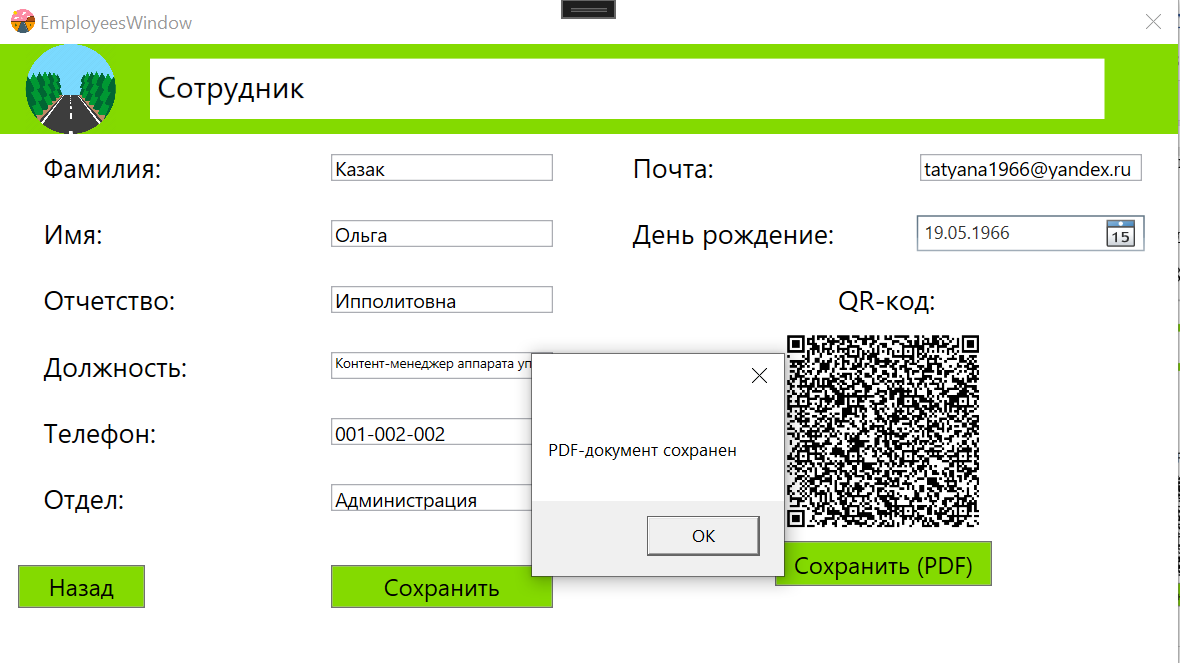


Рисунок 4.11.18 – Результат тестирования

В папке с проектом появился документ с QR-кодом.



Рисунок 4.11.19 – Результат тестирования

При открытии документа и считывания QR-кода с телефона отображается информация о сотруднике (Рисунок 4.11.20). Сравним данные с вводимых полей и получаем результат – данные идентичны.

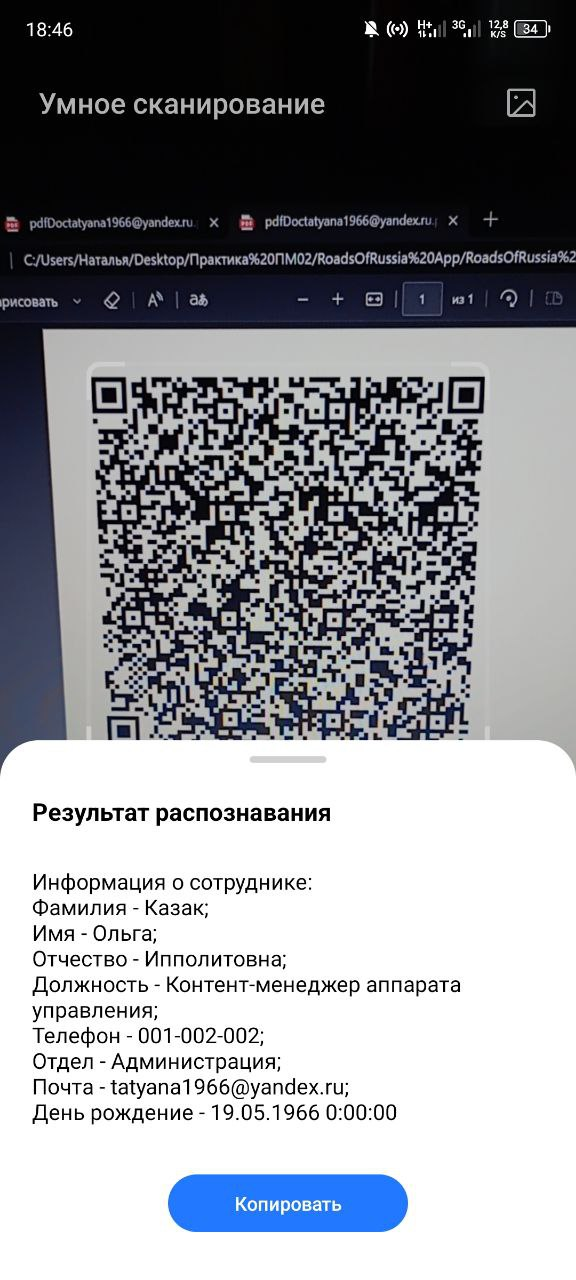


Рисунок 4.11.20 – Результат тестирования

**Тестирование интерфейса.**

*Удобство использования.*

Интерфейс прост в использовании и навигации.

*Эстетичность.*

Интерфейс минималистичен.

*Полнота.*

Интерфейс охватывает все необходимые функции и возможности.

*Эффективность.*

Интерфейс работает с приемлемой скоростью и временем отклика.

Предусмотрены ограничения на минимальные размеры окон, при этом можно увеличить размер окна (Рисунок 4.11.21).

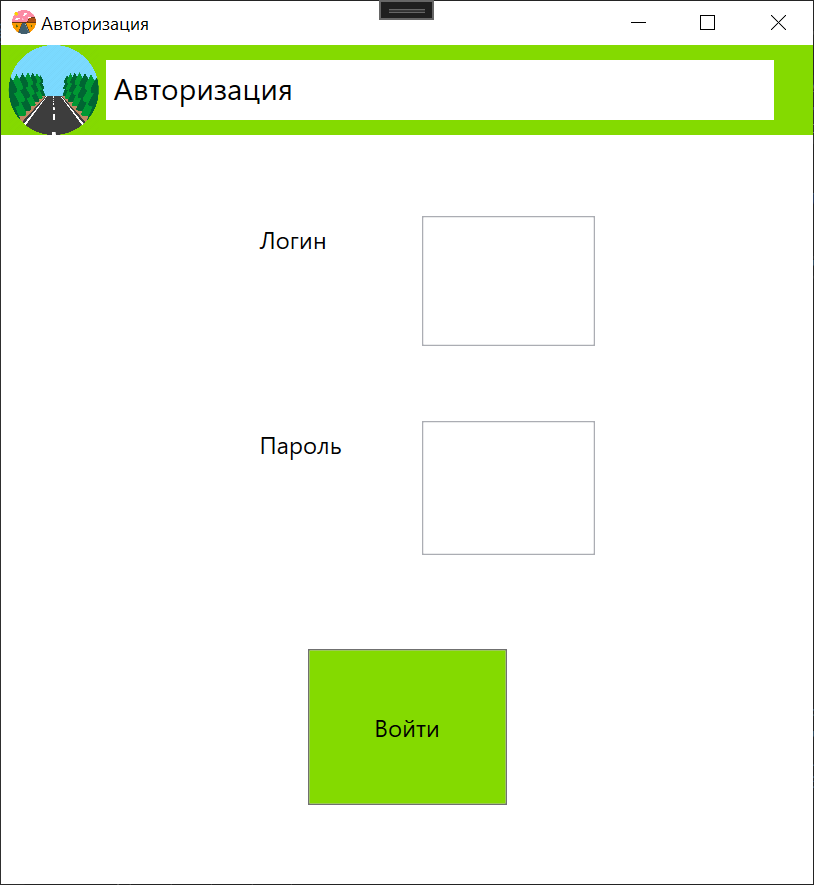


Рисунок 4.11.21 – Изменение размеров окна

При изменении размера окна происходит пропорциональное изменение размеров элементов, расположенных на форме.

На остальных окнах отсутствует свойство изменения размеров окна.

# **Выявлены ошибки системных компонентов (при наличии)**

В ходе анализа программного продукта, реализованного в рамках производственной практики, были выявлены потенциальные ошибки системных компонентов, которые могут повлиять на стабильность и безопасность приложения.

*Неэффективное управление памятью:*

Приложение работает с большим количеством данных о сотрудниках, что может привести к неэффективному управлению памятью, утечкам памяти и замедлению работы приложения.

*Проблемы с безопасностью:*

Отсутствие шифрования данных в процессе хранения и передачи.

*Недокументированный ко*д:

Отсутствие достаточной документации кода, что делает трудным его понимание и поддержание.

В процессе анализа программного продукта были выявлены потенциальные ошибки системных компонентов. Рекомендации по устранению этих ошибок должны быть рассмотрены в рамках дальнейшей разработки приложения, чтобы обеспечить его стабильность, безопасность и удобство использования.

# **Заполнены протоколы тестирования**

ПРОТОКОЛ ТЕСТИРОВАНИЯ

RoadsOfRussia App на ОС «Windows 10»

1. Общая информация

Дата тестирования: 03.07.2024.

Тестировал: Коротких Наталья.

2. Результаты проведённых испытаний

Тест 01 – Функция «авторизация».

Цель данного теста: протестировать функцию авторизации.

Таблица 6 – Тест 01

|  |  |  |  |
| --- | --- | --- | --- |
| # | Выполняемое действие | Ожидаемый результат | Шаг |
| 1 | Авторизация в системе с валидными учетными данными. | Успешная авторизация.  Переход на следующее окно. | ПРОЙДЕН |

Тест 02 – Проверка поиска по полному совпадению фамилии.

Цель данного теста: проверить функцию поиска.

Таблица 7 – Тест 02

|  |  |  |  |
| --- | --- | --- | --- |
| # | Выполняемое действие | Ожидаемый результат | Шаг |
| 1 | Ввод фамилии «Иванов» в строку поиска. | Отображается список сотрудников с фамилией «Иванов». | ПРОЙДЕН |

Тест 03 – Проверка фильтрации по выбранному филиалу.

Цель данного теста: проверить функцию фильтрации.

Таблица 8 – Тест 03

|  |  |  |  |
| --- | --- | --- | --- |
| # | Выполняемое действие | Ожидаемый результат | Шаг |
| 1 | Выбор области «Кировская область». | Отображается список сотрудников, работающих в «Кировской области» | ПРОЙДЕН |
| 2 | Выбор филиала «Кировский филиал №1». | Отображается список сотрудников, работающих в «Кировском филиале №1» | ПРОЙДЕН |

Тест 04 – Проверка редактирования данных о сотруднике.

Цель данного теста: проверить функцию редактирования данных о сотруднике.

Таблица 9 – Тест 04

|  |  |  |  |
| --- | --- | --- | --- |
| # | Выполняемое действие | Ожидаемый результат | Шаг |
| 1 | Выбор записи о сотруднике из списка сотрудника. | Открывается окно редактирования данных о сотруднике. | ПРОЙДЕН |
| 2 | Изменение данных о сотруднике (изменение почты). | Если данные внесены не корректно, то должны быть соответствующее сообщение. | ПРОВАЛЕН |
| 3 | Сохранение изменений. | Сообщение о том, что изменения успешно сохранены. | ПРОЙДЕН |
|  | **ИТОГОВЫЙ ВЫВОД:** | 2 шага из 3 пройдены, т. е. 67% положительного результата | ПРОЙДЕН |

# **Продемонстрировано знание стандартов кодирования более чем одного языка программирования**

*Языки программирования и технологии:*

* C# – основной язык программирования для реализации логики приложения.
* Visual Studio – среда разработки для создания приложения.
* SQL Server Management Studio – инструмент для управления базой данных SQL Server.
* NET Framework – платформа для разработки приложений.
* MessagingToolkit.QRCode – библиотека для генерации QR-кода.
* iTextSharp – библиотека для создания PDF-документов.

Пример стандарта кодирования языке C# (Рисунок 4.14.1):

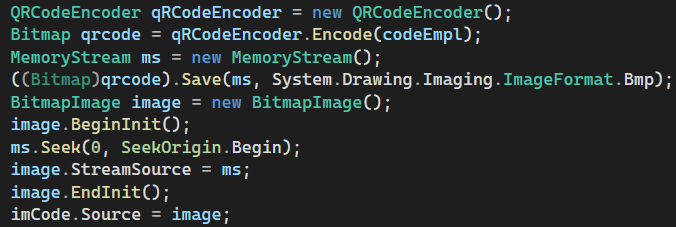


Рисунок 4.14.1 - Пример стандарта кодирования языке C#

Пример кода на C# иллюстрирует реализацию генерации QR-кода.

Пример стандарта кодирования одно из языков программирование – SQL-запросов (Рисунок 4.14.2):

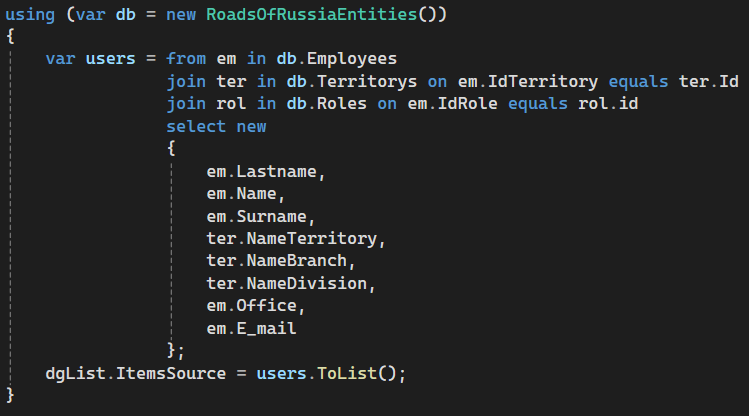


Рисунок 4.14.2 - Пример стандарта кодирования

Этот код выполняет запрос к базе данных, чтобы получить список сотрудников и отобразить его в элементе управления dgList, который, является DataGrid.

# **Выявлены все имеющиеся несоответствия стандартам в предложенном коде**

*Дублирование кода:*

Вызов AllEmp в методах btAllEmployees\_Click и Button\_Click является дублированием.

*Неэффективное использование ресурсов:*

Создание нового контекста базы данных (RoadsOfRussiaEntities) внутри каждого метода, работающего с базой данных, приводит к потере производительности.

*Неудобство использования:*

Метод dgList\_SelectionChanged некорректно использует SelectedIndex для передачи информации в EmployeesWindow.

SelectedIndex может быть некорректным, если в dgList используются группировки или сортировка.

*Неправильное именование:*

terr, brush, devis – слишком короткие и неинформативные. Лучше использовать selectedTerritory, selectedBranch, selectedDivision.

*Неправильное использование переменных:*

terr, brush, devis – переменные доступны глобально, что может привести к ошибкам при одновременном доступе из разных потоков.

Переменные используются неэффективно: вместо того, чтобы каждый раз создавать новые запросы к базе данных, можно использовать уже полученные данные.

*Отсутствие комментариев:*

dgList – не понятно, что это за переменная.

tviTerrLevel0\_Selected, tviTerrLevel1\_Selected, tviTerrLevel2\_Selected, AllEmp – не понятно, что делают методы.

lvNews – не понятно, что это за переменная.

*Потенциальные ошибки:*

MessageBox.Show("Ошибка"+ex.ToString()) – не рекомендуется выводить сообщение об ошибке прямо в MessageBox. Лучше использовать logger для логирования ошибок, а пользователю выводить более дружественное сообщение.

# **РУКОВОДСТВО ОПЕРАТОРА**

1. **Назначение программы**.

Данное приложение предназначено для управления информацией о сотрудниках компании. Оно позволяет:

* Просматривать список сотрудников.
* Поиск сотрудников по ФИО, отделу и почте.
* Фильтровать сотрудников по области, филиалу и отделу.
* Просматривать подробную информацию о сотруднике.
* Изменять данные о сотруднике.
* Сохранять изменения о сотруднике в базе данных.
* Сохранять QR-код сотрудника в PDF-файл.

1. **Условия выполнения программы.**

Для работы с приложением необходим компьютер с установленной операционной системой Windows.

Должна быть установлена и настроена база данных (SQL Server Management Studio), в которой хранятся данные о сотрудниках.

Необходимо иметь учетную запись с логином и паролем для доступа к приложению.

1. **Выполнение программы.**

**3.1 Авторизация**

* Запустите приложение.
* Введите логин и пароль, предоставленные администратором.
* Нажмите кнопку «Войти».

**3.2 Главное окно**

* После успешной авторизации вы перейдете в главное окно приложения.
* В главном окне вы увидите таблицу со списком всех сотрудников.
* В таблице отображается информация о каждом сотруднике, включая – ФИО, Область, Филиал, Отдел, Кабинет, Почта.

**3.3.** **Поиск сотрудника**

* Введите фамилию сотрудника в поле поиска.

**3.4. Фильтрация сотрудников по области**

* Выберите область из списка в фильтре.

**3.5. Фильтрация сотрудников по филиалу**

* Выберите область из списка в фильтре.
* Выберите филиал из выпадающего списка в фильтре.

**3.6. Фильтрация сотрудников по отделу**

* Выберите область из списка в фильтре.
* Выберите филиал из выпадающего списка в фильтре.
* Выберите отдел из выпадающего списка в фильтре.

**3.7. Просмотр информации о сотруднике**

* Нажмите на запись с сотрудником в таблице, чтобы открыть окно с его информацией.

**3.8. Изменение данных о сотруднике**

* Введите новые данные в соответствующие поля в окне с информацией о сотруднике.
* Нажмите кнопку «Сохранить» для сохранения изменений.

**3.9. Сохранение QR-кода в PDF**

* В окне с информацией о сотруднике нажмите кнопку «Сохранить (PDF)».

**3.10. Выход из системы**

• Чтобы выйти из системы, нажмите кнопку «Выход» на главном окне.

1. **Сообщения оператору.**

* Ошибка авторизации:

При вводе неверных логина или пароля появится сообщение об ошибке.

* Ошибка сохранения данных:

Если произошла ошибка при сохранении изменений, появится сообщение об ошибке.

* Успешное сохранение данных:

После успешного сохранения изменений появится сообщение об успешном завершении операции.

* Ошибка создания PDF-файла:

При возникновении ошибки при создании PDF-файла появится сообщение об ошибке.

# **ЗАКЛЮЧЕНИЕ**

В период с 17 июня по 6 июля 2024 года я проходила производственную практику по специальности 09.02.07 Информационные системы и программирование по модулю ПМ.02 Интеграция программных модулей в КОГПОБУ «Слободском колледже педагогики и социальных отношений».

Производственная практика в КОГПОБУ «Слободском колледже педагогики и социальных отношений» позволила мне углубить и закрепить теоретические знания, полученные в учебном заведении. В ходе практики я проследила связь теоретических основ с практическим их применением в сфере информационных систем и программирования.

В начале практики я ознакомилась с технической документацией на используемое в колледже программное обеспечение, что позволило мне получить представление о существующей системе и ее особенностях.

Во время практики я успешно справилась с рядом задач, которые позволили мне применить и развить мои профессиональные навыки:

* Разработка программного решения:

Опыт позволил мне применить знания об алгоритмах, структурах данных и языках программирования для создания функционального приложения.

* Интеграция нового программного модуля:

Я успешно интегрировала новый программный модуль в существующую систему. Эта задача потребовала от меня глубокого понимания архитектуры системы, а также навыков работы с различными форматами данных.

* Тестирование и отладка проекта:

Я провела тестирование и отладку разработанных решений, что позволило выявить и устранить ошибки. Это помогло мне улучшить качество кода и убедиться в его корректной работе.

В ходе практики я научилась применять свои знания и навыки на практике, приобрела ценный опыт работы в реальной производственной среде и укрепила свои профессиональные компетенции.

Полученные во время практики знания и навыки несомненно будут полезны в моей будущей профессиональной деятельности в качестве специалиста по информационным системам и программированию.

Практика помогла мне подготовиться к работе в ИТ-сфере и повысила мою уверенность в своих силах.

# **ПРИЛОЖЕНИЯ К ОТЧЕТУ: ОТЧЕТ В ЭЛЕКТРОННОМ ВИДЕ, ПРЕЗЕНТАЦИЯ ДЛЯ ВЫСТУПЛЕНИЯ И ДР. МАТЕРИАЛЫ**

Ссылка на репозиторий на GitHub по учебной практике – <https://github.com/Natasha975/Industrial-practice-PM.02.git>.

На данном репозитории располагается разработанное решение, скрипт базы данных, документы по производственной практике в электронном виде, а также презентация, подготовленная для защиты результатов производственной практики. Все материалы представлены в электронном виде, что позволяет удобно изучить и проанализировать результаты моей работы.