**МИНИСТЕРСТВО ПРОМЫШЛЕННОСТИ И ТОРГОВЛИ**

**ТВЕРСКОЙ ОБЛАСТИ**

**Государственное бюджетное профессиональное образовательное учреждение**

**«Вышневолоцкий колледж»**

**ОТЧЕТ**

На тему: **«»**

По специальности: 09.02.07 «Информационные системы и программирование»

По ПМ.02 «»

Выполнил: студент группы П-46

Пантелеев Всеволод Владимирович

Руководитель проекта:

Яковлева Жанна Сергеевна

г. Вышний Волочек

2024 г.

**СОДЕРЖАНИЕ**

[1. ТЕХНИКА БЕЗОПАСНОСТИ В КАБИНЕТЕ ИНФОРМАТИКИ 3](#_Toc157767875)

[2. ТЕХНИЧЕСКОЕ ЗАДАНИЕ 6](#_Toc157767876)

[3. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 21](#_Toc157767877)

[4. PROJECT. РАЗРАБОТАТЬ ПРОЕКТ ПО ИНДИВИДУАЛЬНОМУ ЗАДАНИЮ 27](#_Toc157767878)

[5. ДИАГРАММА ERD 27](#_Toc157767879)

[6. ДИАГРАММА ВЗАИМОДЕЙСТВИЯ 28](#_Toc157767880)

[7. ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ 28](#_Toc157767881)

[8. ЛАБОРАТОРНАЯ РАБОТА 1 29](#_Toc157767882)

[9. ЛАБОРАТОРНАЯ РАБОТА 2 29](#_Toc157767883)

[10. ЛАБОРАТОРНАЯ РАБОТА 3 30](#_Toc157767884)

[11. ЛАБОРАТОРНАЯ РАБОТА 4 31](#_Toc157767885)

[12. ЛАБОРАТОРНАЯ РАБОТА 5 32](#_Toc157767886)

[13. ЗАПРОС НА СОЗДАНИЕ БД И ФОРМИРОВАНИЕ SQL ДИАГРАММЫ 32](#_Toc157767887)

[14. ERD VISIO 34](#_Toc157767888)

[15. ЗАПРОС НА СОЗДАНИЕ БД, КОТОРАЯ СООТВЕТСТВУЕТ СХЕМЕ. ПО ВАРИАНТАМ 35](#_Toc157767889)

[16. СОЗДАНИЕ ПРИЛОЖЕНИЯ 41](#_Toc157767890)

[17. АЛГОРИТМ В ВИДЕ БЛОК-СХЕМЫ (ГОСТ 19.701) 55](#_Toc157767891)

[18. АЛГОРИТМ В ВИДЕ ТАБЛИЦ (ГОСТ 2.105) 56](#_Toc157767892)

[19. АЛГОРИТМ В ВИДЕ ТЕКСТОВОГО ОПИСАНИЯ (ГОСТ 24.301) 57](#_Toc157767893)

[20. ПРОТОТИП ПРОГРАММЫ 58](#_Toc157767894)

[21. ER-ДИАГРАММА (ПРОКАТ АВТОМОБИЛЕЙ) 60](#_Toc157767895)

[22. АЛГОРИТМ В ВИДЕ БЛОК-СХЕМЫ (ПРОКАТ АВТОМОБИЛЕЙ) 60](#_Toc157767896)

[23. АЛГОРИТМ В ВИДЕ ТАБЛИЦ (ПРОКАТ АВТОМОБИЛЕЙ) 61](#_Toc157767897)

[24. АЛГОРИТМ В ВИДЕ ТЕКСТОВОГО ОПИСАНИЯ (ПРОКАТ АВТОМОБИЛЕЙ) 62](#_Toc157767898)

[25. БАЗА ДАННЫХ (ПРОКАТ АВТОМОБИЛЕЙ) 63](#_Toc157767899)

[26. ФОРМА АВТОРИЗАЦИИ (ПРОКАТ АВТОМОБИЛЕЙ) 68](#_Toc157767900)

[27. РУКОВОДСТВО СИСТЕМНОМУ ПРОГРАММИСТУ (ГОСТ 19.503-79) 71](#_Toc157767901)

**1. ТЕХНИКА БЕЗОПАСНОСТИ В КАБИНЕТЕ ИНФОРМАТИКИ**

К работе в кабинете информатики допускаются только учащиеся и преподаватели, прошедшие инструктаж по технике безопасности, соблюдающие указания преподавателя, расписавшиеся в журнале регистрации инструктажа.

Необходимо неукоснительно соблюдать правила по технике безопасности, т.к. нарушение этих правил может привести к поражению электрическим током, вызвать возгорание и навредить вашему здоровью.

При эксплуатации оборудования необходимо остерегаться:

* Поражения электрическим током;
* Механических повреждений, травм.

Требования безопасности перед началом работы:

* Запрещено входить в кабинет в верхней одежде, головных уборах, с громоздкими предметами и едой;
* Запрещено входить в кабинет информатики в грязной обуви без бахил или без сменной обуви;
* Запрещается шуметь, громко разговаривать и отвлекать других учащихся;
* Запрещено бегать и прыгать, самовольно передвигаться по кабинету;
* Перед началом занятий все личные мобильные устройства учащихся (телефон, плеер и т.п.) должны быть выключены;
* Разрешается работать только на том компьютере, который выделен на занятие;
* Перед началом работы учащийся обязан осмотреть рабочее место и свой компьютер на предмет отсутствия видимых повреждений оборудования;
* Запрещается выключать или включать оборудование без разрешения преподавателя;
* Напряжение в сети кабинета включается и выключается только преподавателем.

Требования безопасности во время работы:

* С техникой обращаться бережно: не стучать по мониторам, не стучать мышкой о стол, не стучать по клавишам клавиатуры;
* При возникновении неполадок: появлении изменений в функционировании аппаратуры, самопроизвольного её отключения необходимо немедленно прекратить работу и сообщить об этом преподавателю;
* Не пытаться исправить неполадки в оборудовании самостоятельно;
* Выполнять за компьютером только те действия, которые говорит преподаватель;
* Контролировать расстояние до экрана и правильную осанку;
* Не допускать работы на максимальной яркости экрана дисплея;
* В случае возникновения нештатных ситуаций сохранять спокойствие и чётко следовать указаниям преподавателя.

Запрещается:

* Эксплуатировать неисправную технику;
* При включённом напряжении сети отключать, подключать кабели, соединяющие различные устройства компьютера;
* Работать с открытыми кожухами устройств компьютера;
* Касаться экрана дисплея, тыльной стороны дисплея, разъёмов, соединительных кабелей, токоведущих частей аппаратуры;
* Касаться автоматов защиты, пускателей, устройств сигнализации;
* Во время работы касаться труб, батарей;
* Самостоятельно устранять неисправность работы клавиатуры;
* Нажимать на клавиши с усилием или допускать резкие удары;
* Пользоваться каким-либо предметом при нажатии на клавиши;
* Передвигать системный блок, дисплей или стол, на котором они стоят;
* Загромождать проходы в кабинете сумками, портфелями, стульями;
* Брать сумки, портфели за рабочее место у компьютера;
* Брать с собой в класс верхнюю одежду и загромождать ею кабинет;
* Быстро передвигаться по кабинету;
* Класть какие-либо предметы на системный блок, дисплей, клавиатуру;
* Работать грязными, влажными руками, во влажной одежде;
* Работать при недостаточном освещении;
* Работать за дисплеем дольше положенного времени.

Запрещается без разрешения преподавателя:

* Включать и выключать компьютер, дисплей и другое оборудование;
* Использовать различные носители информации (дискеты, диски, флешки);
* Подключать кабели, разъёмы и другую аппаратуру к компьютеру;
* Брать со стола преподавателя дискеты, аппаратуру, документацию и другие предметы;
* Пользоваться преподавательским компьютером.

Требования безопасности по окончанию работы:

* По окончании работы дождаться пока преподаватель подойдёт и проверит состояние оборудования, сдать работу, если она выполнялась;
* Медленно встать, собрать свои вещи и тихо выйти из класса, чтобы не мешать другим учащимся.

Ответственность за нарушение правил техники безопасности:

* При нарушении техники безопасности учащемуся будет объявлен выговор, взыскание вплоть до отстранения от работы за оборудованием;
* При регулярных нарушениях техники безопасности учащийся будет отстранён от занятий информатики вплоть до исключения из учебного заведения;

**2. ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**Программа «Моя записная книжка»**

**Техническое задание**

**Листов «14»**

г. Вышний Волочек

2024 г.

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ………………………………………………………………..………9

1. Основания для разработки………………………………………...…………11

2. Назначение разработки……………………………………………………….13

3. Требования к программе или программному изделию……………………..14

3.1. Требования к функциональным характеристикам………………………..14

3.2. Требования к надежности…………………………………………………..14

3.3. Условия эксплуатации …………………………………………………..…15

3.4. Требования к составу и параметрам технических средств……...……….16

3.5. Требования к информационной и программной совместимости…….…16

3.6. Требования к маркировке и упаковке………………………………..……16

3.7. Требования к транспортированию и хранению…………....……...………17

3.8. Специальные требования ………………………………………..…………17

4. Требования к программной документации…………………………….…...18

5. Технико-экономические показатели…………………………………..…….19

6. Стадии и этапы разработки…………………………………….…………….20

7. Порядок контроля и приемки………………………………………....……...21

**ВВЕДЕНИЕ**

Настоящее техническое задание распространяется на разработку программы «Моя записная книжка», используемой для работы с произвольным числом записей, поиска записи по какому-либо признаку (например, по фамилии, дате рождения или номеру телефона), добавления и удаления записей, сортировки по разным полям.

Хранение данных организовано с применением контейнерных классов, таких как set или multimap. Программа предназначена для работы на операционных системах семейства Windows, начиная с версии XP, и использует язык программирования C#. Разработку предполагается вести в среде Microsoft Visual Studio.

Целью разработки является создание удобного инструмента для управления контактной информацией, который позволит пользователям быстро находить нужные записи, добавлять новые и удалять ненужные.

В состав программы должны входить следующие модули и компоненты:

* Главное окно, содержащее список контактов;
* Форма добавления нового контакта с полями для ввода информации;
* Форма редактирования существующего контакта;
* Форма удаления контакта;

Все формы должны иметь интуитивно понятный интерфейс, не требующий от пользователя специальных знаний.

Разработку предполагается разбить на следующие этапы:

* Анализ требований и составление технического задания;
* Проектирование архитектуры программы;
* Разработка программного кода;
* Тестирование и исправление ошибок;
* Внедрение и сопровождение.

Контроль качества разработки будет осуществляться на всех этапах, включая тестирование функциональности, интеграцию с другими компонентами и проверку соответствия требованиям.

Цели:

* Создание удобного и функционального инструмента для работы с контактной информацией.
* Обеспечение быстрого поиска нужных записей, добавления новых и удаления ненужных.
* Разработка интуитивно понятного интерфейса, не требующего от пользователя специальных знаний.

Задачи:

* Проанализировать требования и составить техническое задание.
* Спроектировать архитектуру программы.
* Реализовать программный код, отвечающий требованиям технического задания.
* Провести тестирование программы, выявить и исправить возможные ошибки.
* Внести коррективы в программу по результатам тестирования.
* Обеспечить внедрение и сопровождение программы после завершения разработки.

Программа «Моя записная книжка» является актуальной, поскольку она предоставляет пользователям удобный инструмент для работы с контактной информацией, позволяя им быстро находить нужные записи, добавлять новые и удалять ненужные, а также сортировать данные по различным полям. Это может быть особенно полезно для людей, имеющих большое количество контактов или записей, которые нужно организовать и управлять ими эффективно. Кроме того, программа имеет интуитивно понятный интерфейс, который не требует от пользователя специальных знаний, что делает ее доступной для широкого круга пользователей.

**1. Основания для разработки**

Программа «Моя записная книжка» разрабатывается в соответствии со следующими принципами:

* Простота и удобство использования: приложение должно быть интуитивно понятным и легким в использовании для широкого круга пользователей.
* Функциональность: приложение должно предоставлять широкий спектр функций для управления контактной информацией, таких как добавление, редактирование, поиск и сортировка контактов.
* Безопасность: все данные, хранящиеся в приложении, должны быть защищены от несанкционированного доступа и потери.
* Интеграция с другими сервисами: приложение должно поддерживать интеграцию с другими популярными сервисами, такими как социальные сети, мессенджеры и почтовые клиенты, для быстрого и удобного обмена контактной информацией.
* Персонализация: пользователи должны иметь возможность настроить внешний вид приложения и его функционал в соответствии со своими предпочтениями.
* Поддержка различных устройств: приложение должно работать на различных типах устройств, включая смартфоны, планшеты и компьютеры.
* Регулярное обновление и улучшение: приложение должно постоянно обновляться и улучшаться, чтобы оставаться актуальным и востребованным на рынке.

Перечень оснований для разработки:

* Удовлетворение спроса пользователей на удобные и функциональные приложения для управления контактной информацией.
* Расширение аудитории пользователей и привлечение новых клиентов для компании-разработчика.
* Создание конкурентоспособного продукта на рынке приложений для управления контактами.
* Разработка приложения с уникальным функционалом, которое может выделиться среди других подобных приложений.
* Возможность дальнейшей монетизации приложения через продажу рекламы, дополнительных функций или подписки.
* Развитие и продвижение бренда компании-разработчика через качественный и популярный продукт.
* Тестирование новых технологий и возможностей для улучшения функциональности приложения.
* Обучение и развитие команды разработчиков через реализацию сложного и интересного проекта.
* Возможность интеграции приложения с другими продуктами компании для создания комплексной экосистемы.
* Улучшение качества жизни пользователей благодаря оптимизации процессов управления контактной информацией.

**2. Назначение разработки**

Основное назначение программы «Моя записная книжка» заключается в предоставлении пользователям удобного инструмента для работы с контактной информацией, позволяющего им быстро находить нужные записи, добавлять новые и удалять ненужные, а также сортировать данные по различным полям. Кроме того, приложение может иметь дополнительные функции, такие как напоминания о днях рождения контактов, синхронизация данных с другими устройствами, резервное копирование и восстановление данных, а также поддержка различных форматов файлов для импорта и экспорта контактной информации.

Основной функционал программы «Моя записная книжка» включает в себя:

* Добавление, редактирование и удаление контактов;
* Поиск контактов по имени, фамилии, номеру телефона или адресу электронной почты;
* Сортировка контактов по различным полям;
* Возможность создания различных категорий контактов и привязки контактов к этим категориям;
* Импорт и экспорт контактов в различных форматах файлов;
* Напоминания о днях рождениях контактов и других важных событиях;
* Синхронизация контактов с другими устройствами и учетными записями;
* Резервное копирование и восстановление контактов на случай потери или повреждения данных;
* Защита данных с помощью пароля и шифрования;
* Настройка внешнего вида приложения.

**3. Требования к программе или программному изделию**

**3.1. Требования к функциональным характеристикам**

Перечень требований:

* Добавление, редактирование и удаление контактов.
* Поиск контактов по различным параметрам (имя, фамилия, номер телефона, адрес электронной почты и т.д.).
* Сортировка контактов по разным полям (алфавитная, по дате добавления и т.д.).
* Создание различных категорий контактов и привязка контактов к ним.
* Импорт и экспорт контактов в разных форматах файлов (.csv, .xlsx, .vcf и другие).
* Напоминания о днях рождениях и других важных датах.
* Синхронизация контактов с другими устройствами.
* Резервное копирование контактов на случай потери данных.
* Защита данных паролем и шифрованием.
* Настройка внешнего вида приложения.

**3.2. Требования к надежности**

Перечень требований:

* Приложение должно быть стабильным и надежным, без сбоев и ошибок.
* Все данные должны быть сохранены и защищены от потери или повреждения.
* Необходимо предусмотреть возможность восстановления данных в случае возникновения проблем.
* Программа должна работать на разных операционных системах (Windows, MacOS, iOS, Android).
* Для использования программы требуется подключение к интернету.
* Программа должна быть доступной для загрузки на сайте разработчика и в магазинах приложений.
* Интерфейс программы должен быть простым и понятным для пользователей разного уровня подготовки.
* В программе должен быть предусмотрен функционал для добавления, редактирования, удаления контактов, а также их поиска.
* Особые требования к техническим средствам не предъявляются, программа должна работать на большинстве современных устройств с достаточным объемом памяти и производительностью.
* Требуется доступ к интернету для синхронизации контактов и использования дополнительных возможностей приложения.
* Программа должна быть совместимой с другими приложениями и сервисами для обмена контактной информацией (почтовые клиенты, календари и т. д.).
* В качестве дополнительных опций можно предложить поддержку нескольких языков, использование тем оформления, настройку шрифтов, удобство использования для людей с ограниченными возможностями.

**3.3. Условия эксплуатации**

Условия эксплуатации:

* Программа предназначена для работы на компьютерах и мобильных устройствах под управлением операционных систем Windows, macOS, iOS и Android.
* Для использования программы необходимо иметь подключение к интернету.
* Программа должна быть доступна для загрузки на официальном сайте разработчика и в магазинах приложений App Store и Google Play.
* Программа должна иметь простой и интуитивно понятный интерфейс, доступный для пользователей с различным уровнем подготовки.
* Программа должна обеспечивать возможность добавления, редактирования и удаления контактов, а также поиска по ним.

**3.4. Требования к составу и параметрам технических средств**

Программа «Моя записная книжка» не предъявляет особых требований к техническим средствам. Она может работать на большинстве современных компьютеров и мобильных устройств с достаточной производительностью и объемом памяти. Необходим доступ в интернет для синхронизации контактов и использования дополнительных функций приложения.

**3.5. Требования к информационной и программной совместимости**

Программа «Моя записная книжка» должна быть совместима с большинством современных операционных систем (Windows, macOS, iOS, Android) и мобильных устройств. Она также должна работать с популярными почтовыми клиентами, календарями и другими приложениями для обмена контактной информацией. В качестве дополнительных требований можно указать поддержку нескольких языков интерфейса, возможность использования тем оформления и настройку шрифтов.

**3.6. Требования к маркировке и упаковке**

Программа «Моя записная книжка» не имеет физической упаковки и маркировки, так как является программным обеспечением для электронных устройств. Программа распространяется в виде установочных файлов или приложений в магазинах цифровой дистрибуции, таких как App Store, Google Play и других.

**3.7. Требования к транспортированию и хранению**

Программа «Моя записная книжка» может быть транспортирована на любых носителях информации, таких как USB-флешки, оптические диски и другие. Для хранения программы не требуется особых условий, достаточно обычного компьютера или мобильного устройства с доступом в интернет.

**3.8. Специальные требования**

К специальным требованиям можно отнести:

* Поддержка нескольких языков.
* Возможность использования тем оформления.
* Настройка шрифтов.
* Удобство использования для людей с ограниченными возможностями.

**4. Требования к программной документации**

Требования к программной документации:

* Документация должна содержать описание программы, ее функциональности и возможностей, а также руководство пользователя.
* Документация должна быть написана на понятном и доступном языке, без использования сложной терминологии.
* Документация должна быть актуальной и регулярно обновляться с учетом изменений в программе.
* Документация должна включать информацию о том, как использовать программу, как добавлять и редактировать записи, как искать информацию и т.д.

**5. Технико-экономические показатели**

Технико-экономические показатели программы включают в себя:

* Затраты на разработку программы: стоимость труда разработчиков, стоимость оборудования и программного обеспечения, стоимость обучения и сертификации специалистов и т. д.
* Затраты на эксплуатацию программы: стоимость поддержки и обновления программного обеспечения, затраты на обучение пользователей, затраты на обслуживание и ремонт оборудования и т. д.
* Экономический эффект от использования программы: увеличение производительности труда, снижение затрат на обработку информации, повышение качества принимаемых решений и т. д.

**6. Стадии и этапы разработки**

Стадии и этапы разработки:

1. Анализ требований и определение целей разработки.
2. Проектирование структуры приложения и его основных функций.
3. Разработка интерфейса пользователя и дизайна приложения.
4. Написание кода приложения на выбранном языке программирования.
5. Тестирование приложения на различных устройствах и платформах.
6. Отладка и исправление ошибок в коде приложения.
7. Подготовка документации по использованию приложения.
8. Публикация приложения в магазинах приложений и его продвижение.
9. Поддержка и обновление приложения после его выпуска.

**7. Порядок контроля и приемки**

Порядок контроля и приемки:

* Контроль требований и спецификации.
* Тестирование функциональности и производительности.
* Проверка соответствия стандартам и нормам.
* Оценка удобства использования и эргономики.
* Аудит безопасности и защиты данных.
* Приемка программы заказчиком и подписание акта выполненных работ.

**3. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

**Платформа «GitHub»**

**Руководство пользователя**

**Листов «4»**

г. Вышний Волочек

2024 г.

**СОДЕРЖАНИЕ**

Введение………………………………………………………………..……...…24

1. Регистрация и создание репозитория…………………………...…………...24

2. Клонирование репозитория.………………………………………………….24

3. Добавление изменений и фиксация коммитов.……………………………..25

4. Отправка изменений на GitHub……………………………………………...25

5. Совместная работа с другими разработчиками……………………………..25

6. Обновление локального репозитория.…………………………..………..…26

7. Отслеживание проблем и запросов на изменения……...……………....….26

**Введение**

GitHub — это веб-платформа для хранения и совместной работы с исходным кодом проектов. Она позволяет разработчикам совместно работать над проектами, отслеживать изменения в коде, управлять версиями, а также проводить код-ревью.

**1. Регистрация и создание репозитория.**

Перейдите на веб-сайт GitHub и нажмите кнопку "Sign Up" (Зарегистрироваться), чтобы создать аккаунт.

После успешной регистрации войдите в свою учетную запись.

Нажмите кнопку "Create a repository" (Создать репозиторий) и введите имя репозитория.

Нажмите кнопку "Create repository" (Создать репозиторий), чтобы завершить создание репозитория.

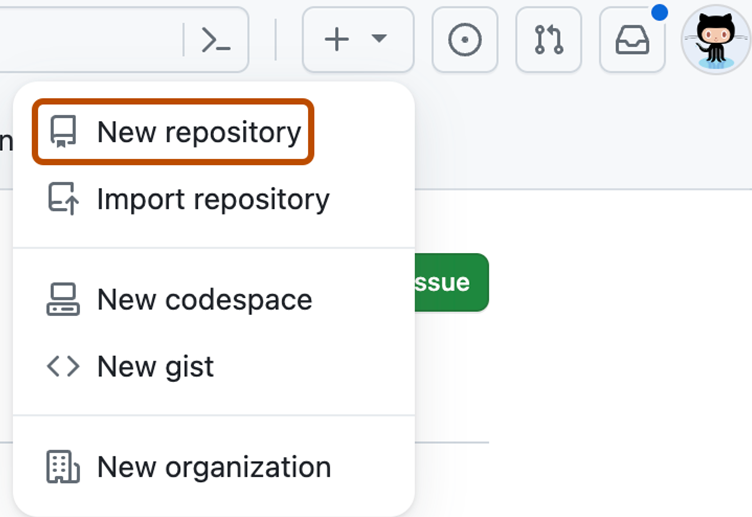


Рис. 1 Новый репозиторий

**2. Клонирование репозитория.**

Чтобы начать работать с репозиторием на локальном компьютере, склонируйте репозиторий с помощью команды git clone <URL репозитория> в командной строке.

Замените <URL репозитория> на URL вашего репозитория на GitHub.

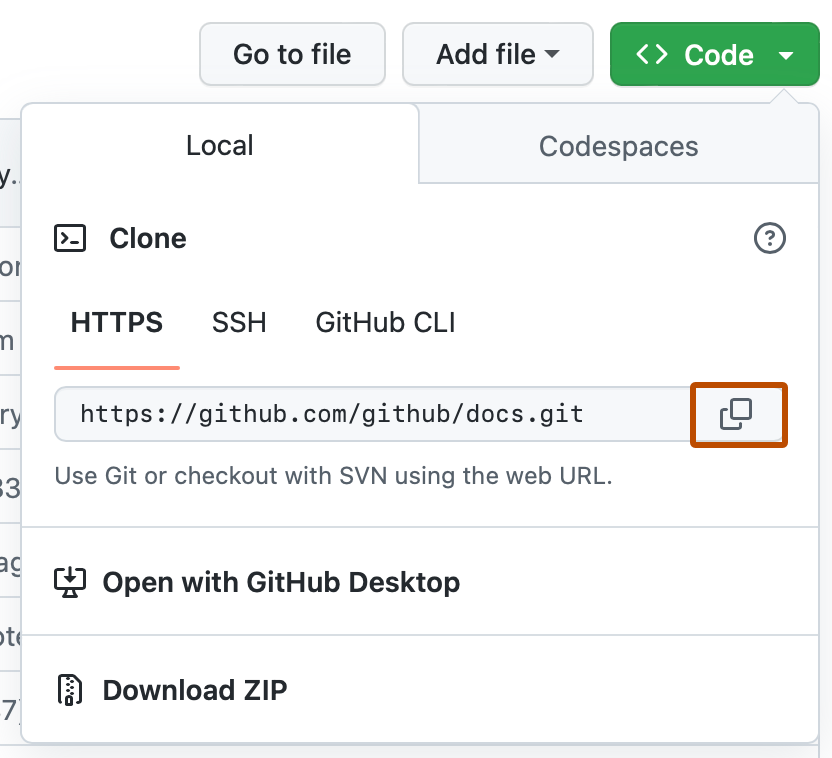


Рис. 2 Клонирование репозитория

**3. Добавление изменений и фиксация коммитов.**

Создайте или измените файлы в склонированном репозитории на вашем компьютере.

Выполните команду git add <файлы> в командной строке, чтобы добавить измененные файлы для коммита.

Выполните команду git commit -m "Описание коммита" для фиксации изменений в коммите.

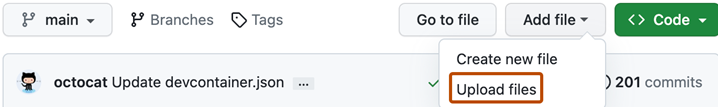


Рис. 3 Добавления файла

**4. Отправка изменений на GitHub.**

Выполните команду git push в командной строке, чтобы отправить коммиты на GitHub.

Ваши изменения теперь будут доступны в вашем репозитории на GitHub.

**5. Совместная работа с другими разработчиками.**

Чтобы пригласить других разработчиков к сотрудничеству, перейдите на вкладку "Settings" (Настройки) вашего репозитория на GitHub.

На странице настроек выберите "Collaborators" (Соавторы) и введите имена других разработчиков.

После приглашения соавторы получат доступ к репозиторию и смогут вносить свои изменения.

**6. Обновление локального репозитория.**

Чтобы получить последние изменения из репозитория на GitHub, выполните команду git pull в командной строке.

Эта команда обновит ваш локальный репозиторий, чтобы совпадать с последней версией на GitHub.

**7. Отслеживание проблем и запросов на изменения.**

Разработчики могут отслеживать проблемы и ошибки в репозитории, создавая "issues" (проблемы) на GitHub.

Чтобы предложить изменения в проекте, разработчики могут создавать "pull requests" (запросы на изменения) на GitHub.

**4. PROJECT. РАЗРАБОТАТЬ ПРОЕКТ ПО ИНДИВИДУАЛЬНОМУ ЗАДАНИЮ**

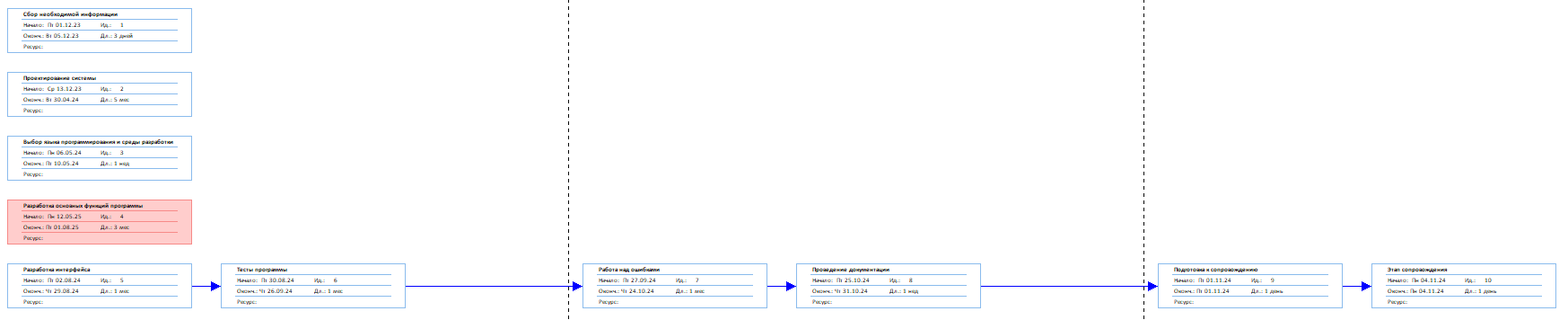


Рис. 4 Проект

**5. ДИАГРАММА ERD**

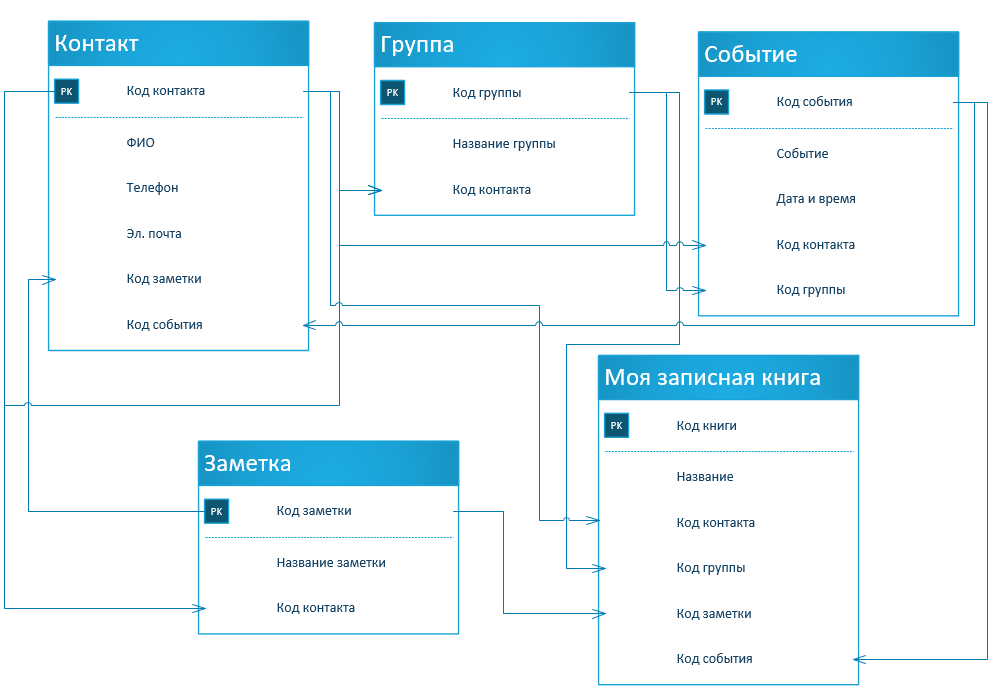


Рис. 5 Диаграмма ERD

**6. ДИАГРАММА ВЗАИМОДЕЙСТВИЯ**

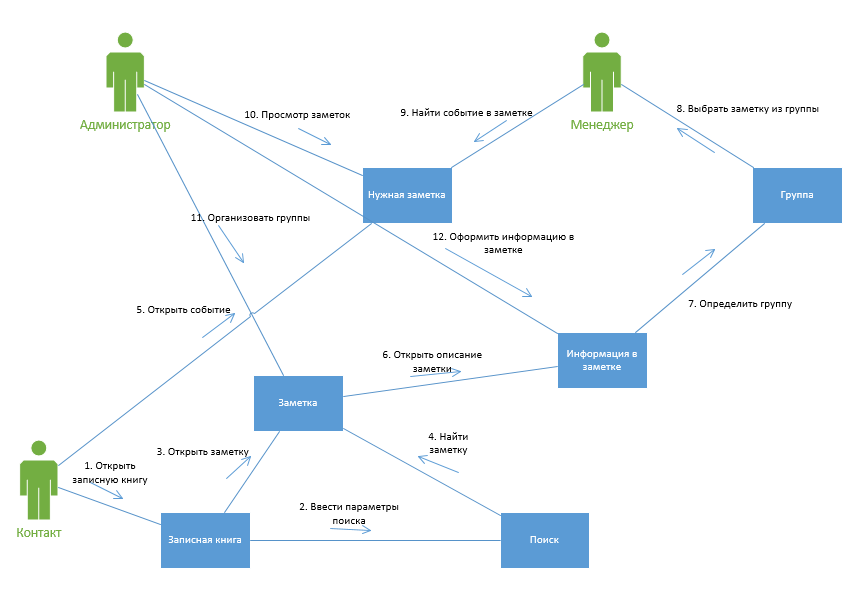


Рис. 6 Диаграмма взаимодействия

**7. ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ**

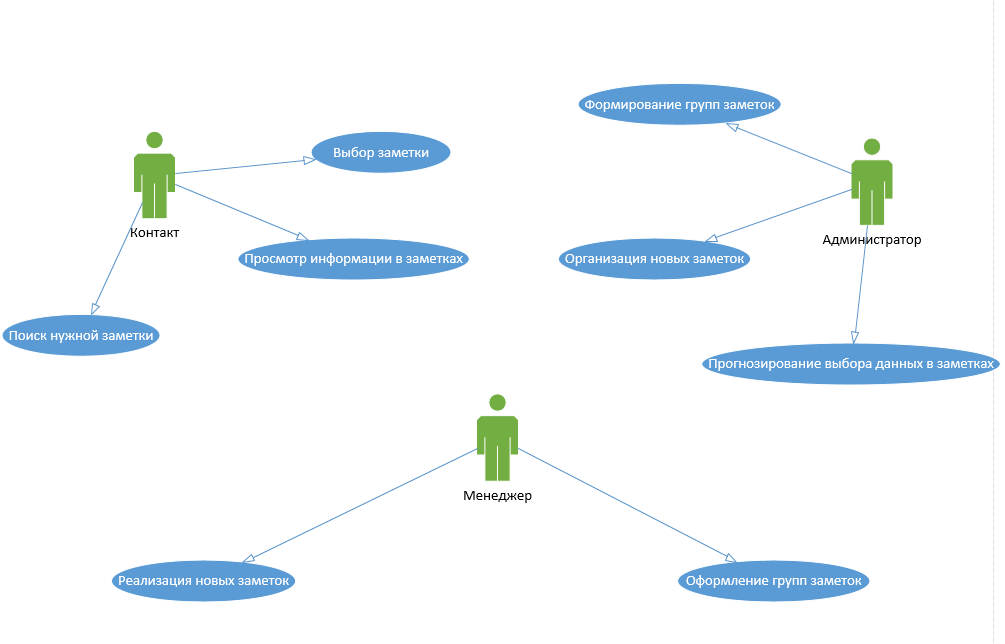


Рис. 7 Диаграмма вариантов использования

**8. ЛАБОРАТОРНАЯ РАБОТА 1**

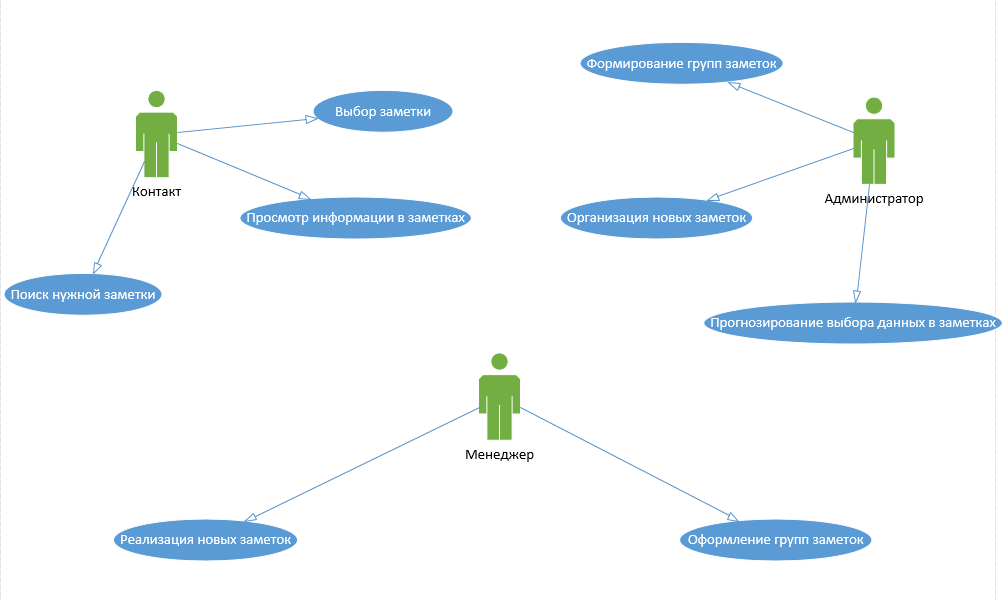


Рис. 8 Лабораторная работа 1

**9.** **ЛАБОРАТОРНАЯ РАБОТА 2**

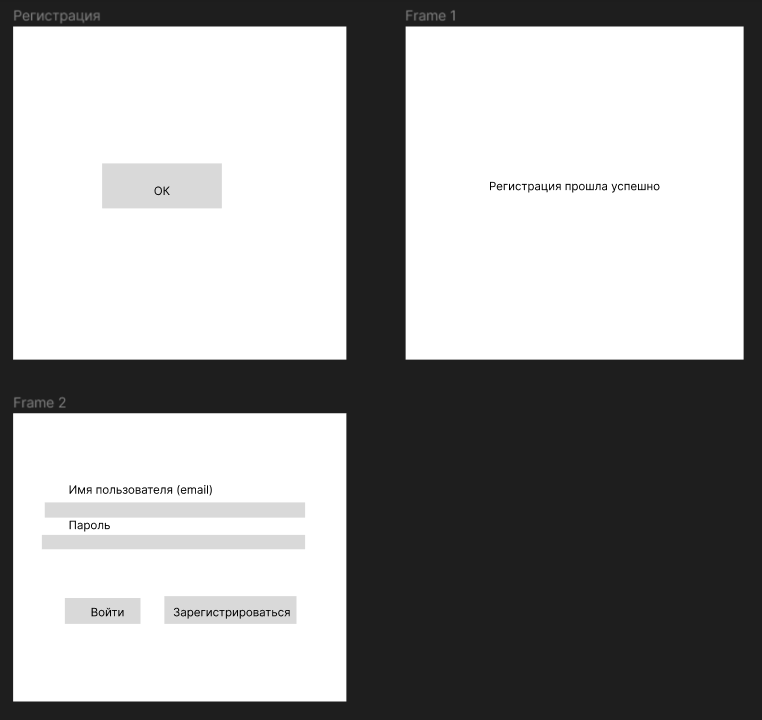


Рис. 9 Лабораторная работа 2

**10. ЛАБОРАТОРНАЯ РАБОТА 3**

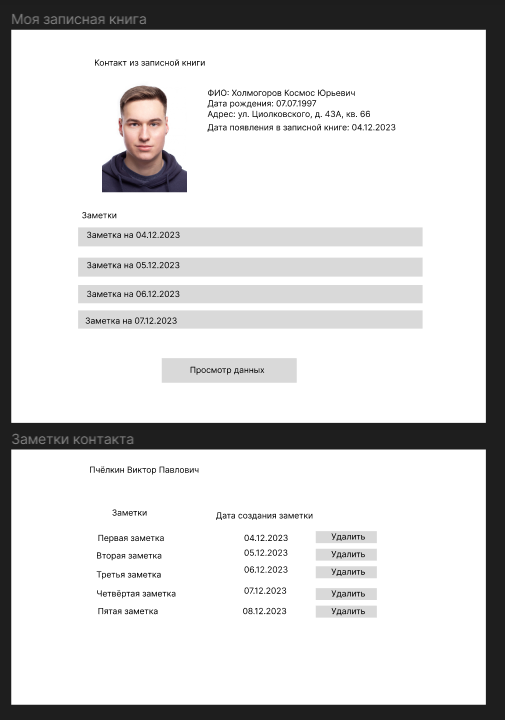


Рис. 10 Лабораторная работа 3

**11. ЛАБОРАТОРНАЯ РАБОТА 4**

**Цель: Тестирование готового прототипа проекта**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **ФИО тестировщика** | **Фрейм для корректировки** | **Замечания** |
| 1 | Белов А.Н. |  | Кнопки являются неработоспособными и не осуществляют переход на другие фреймы |
| 2 | Филатов В.К. |
| 3 | Пчёлкин В.П. |  | Кнопки являются неработоспособными и не осуществляют удаление заметок у пользователя |
| 4 | Холмогоров К.Ю. |
| 5 | Каверин В.Е. |

**12. ЛАБОРАТОРНАЯ РАБОТА 5**

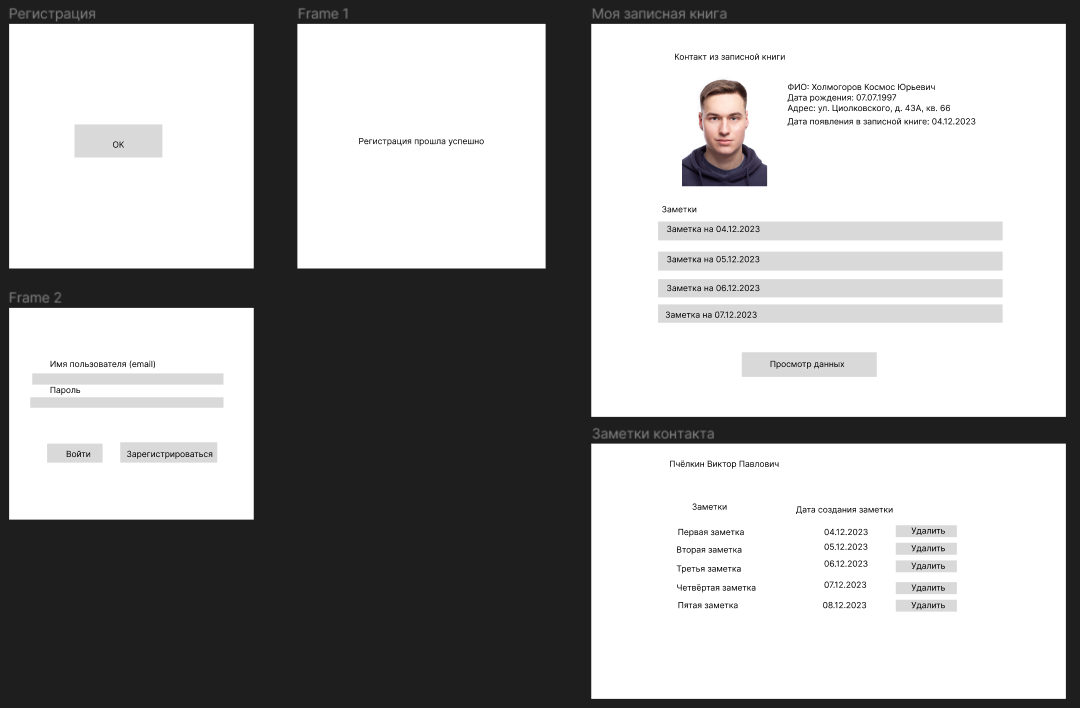


Рис. 11 Лабораторная работа 5

**13. ЗАПРОС НА СОЗДАНИЕ БД И ФОРМИРОВАНИЕ SQL ДИАГРАММЫ**

Код:

CREATE DATABASE BD\_MZK

GO

USE BD\_MZK

CREATE TABLE ZAMETKA

(

Kod\_zametki INT PRIMARY KEY,

Nazvanie\_zametki VARCHAR(250) NOT NULL,

Data\_zametki DATE NOT NULL,

)

CREATE TABLE KONTAKT

(

Kod\_kontakta INT PRIMARY KEY,

FIO VARCHAR(250) NOT NULL,

Telefon VARCHAR(250) NOT NULL,

email VARCHAR(250) NOT NULL,

Kod\_zametki INT FOREIGN KEY REFERENCES ZAMETKA(Kod\_zametki),

)

CREATE TABLE GRUPPA

(

Kod\_gruppi INT PRIMARY KEY,

Nazvanie\_gruppi VARCHAR(250) NOT NULL,

Kod\_kontakta INT FOREIGN KEY REFERENCES KONTAKT(Kod\_kontakta),

)

CREATE TABLE SOBITIE

(

Kod\_sobitiya INT PRIMARY KEY,

Sobitie VARCHAR(250) NOT NULL,

Data DATE NOT NULL,

Kod\_kontakta INT FOREIGN KEY REFERENCES KONTAKT(Kod\_kontakta),

Kod\_gruppi INT FOREIGN KEY REFERENCES GRUPPA(Kod\_gruppi),

)

CREATE TABLE MOYA\_ZAPISNAYA\_KNIGA

(

Kod\_knigi INT PRIMARY KEY,

Nazvanie VARCHAR(250) NOT NULL,

Kod\_kontakta INT FOREIGN KEY REFERENCES KONTAKT(Kod\_kontakta),

Kod\_gruppi INT FOREIGN KEY REFERENCES GRUPPA(Kod\_gruppi),

Kod\_zametki INT FOREIGN KEY REFERENCES ZAMETKA(Kod\_zametki),

Kod\_sobitiya INT FOREIGN KEY REFERENCES SOBITIE(Kod\_sobitiya),

)

**14. ERD VISIO**

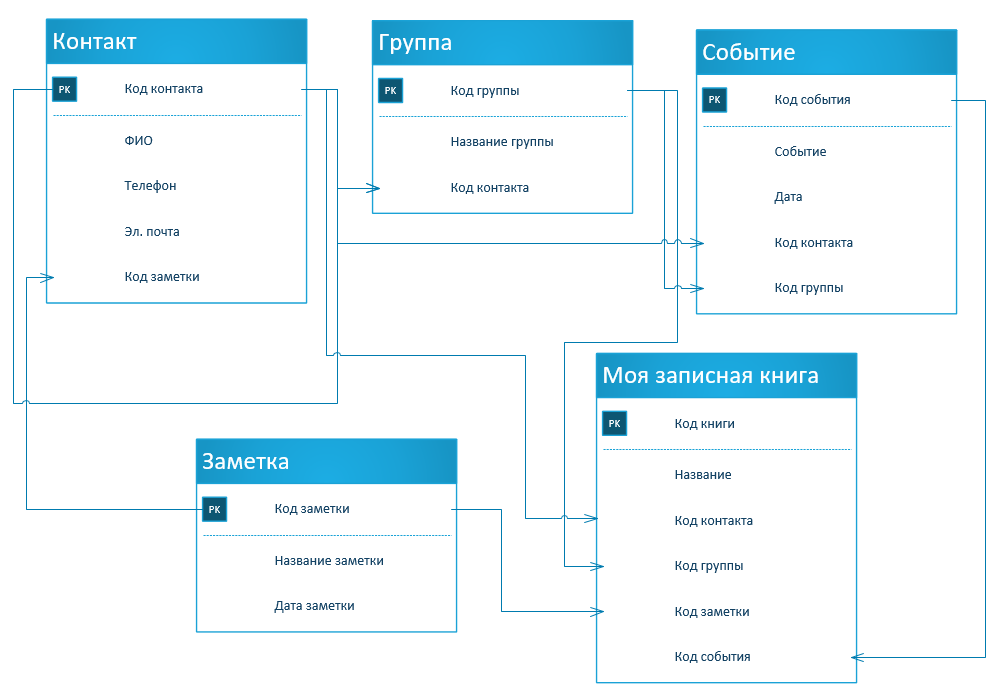


Рис. 12 ERD VISIO

**15. ЗАПРОС НА СОЗДАНИЕ БД, КОТОРАЯ СООТВЕТСТВУЕТ СХЕМЕ. ПО ВАРИАНТАМ**

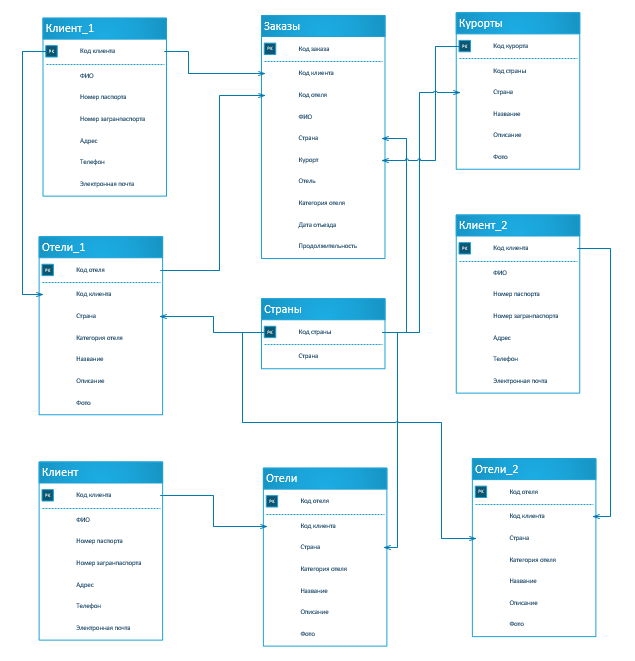


Рис. 13 Диаграмма

Код:

CREATE DATABASE BD\_OTELI

GO

USE BD\_OTELI

CREATE TABLE KLIENT

(

Kod\_klienta INT PRIMARY KEY,

FIO VARCHAR(250) NOT NULL,

Nomer\_pasporta INT NOT NULL,

Nomer\_zagranpasporta INT NOT NULL,

Adres VARCHAR(250) NOT NULL,

Telefon VARCHAR(250) NOT NULL,

Email VARCHAR(250) NOT NULL,

)

CREATE TABLE KLIENT\_1

(

Kod\_klienta INT PRIMARY KEY,

FIO VARCHAR(250) NOT NULL,

Nomer\_pasporta INT NOT NULL,

Nomer\_zagranpasporta INT NOT NULL,

Adres VARCHAR(250) NOT NULL,

Telefon VARCHAR(250) NOT NULL,

Email VARCHAR(250) NOT NULL,

)

CREATE TABLE KLIENT\_2

(

Kod\_klienta INT PRIMARY KEY,

FIO VARCHAR(250) NOT NULL,

Nomer\_pasporta INT NOT NULL,

Nomer\_zagranpasporta INT NOT NULL,

Adres VARCHAR(250) NOT NULL,

Telefon VARCHAR(250) NOT NULL,

Email VARCHAR(250) NOT NULL,

)

CREATE TABLE STRANI

(

Kod\_strani INT PRIMARY KEY,

Strana VARCHAR(250) NOT NULL,

)

CREATE TABLE OTELI

(

Kod\_otelya INT PRIMARY KEY,

Kod\_klienta INT FOREIGN KEY REFERENCES KLIENT(Kod\_klienta),

Strana INT FOREIGN KEY REFERENCES STRANI(Kod\_strani),

Kategoria\_otelya VARCHAR(250) NOT NULL,

Nazvanie VARCHAR(250) NOT NULL,

Opisanie VARCHAR(250) NOT NULL,

Foto VARCHAR(250) NOT NULL,

)

CREATE TABLE OTELI\_1

(

Kod\_otelya INT PRIMARY KEY,

Kod\_klienta INT FOREIGN KEY REFERENCES KLIENT\_1(Kod\_klienta),

Strana INT FOREIGN KEY REFERENCES STRANI(Kod\_strani),

Kategoria\_otelya VARCHAR(250) NOT NULL,

Nazvanie VARCHAR(250) NOT NULL,

Opisanie VARCHAR(250) NOT NULL,

Foto VARCHAR(250) NOT NULL,

)

CREATE TABLE OTELI\_2

(

Kod\_otelya INT PRIMARY KEY,

Kod\_klienta INT FOREIGN KEY REFERENCES KLIENT\_2(Kod\_klienta),

Strana INT FOREIGN KEY REFERENCES STRANI(Kod\_strani),

Kategoria\_otelya VARCHAR(250) NOT NULL,

Nazvanie VARCHAR(250) NOT NULL,

Opisanie VARCHAR(250) NOT NULL,

Foto VARCHAR(250) NOT NULL,

)

CREATE TABLE KURORTI

(

Kod\_kurorta INT PRIMARY KEY,

Kod\_strani INT FOREIGN KEY REFERENCES STRANI(Kod\_strani),

Strana VARCHAR(250) NOT NULL,

Nazvanie VARCHAR(250) NOT NULL,

Opisanie VARCHAR(250) NOT NULL,

Foto VARCHAR(250) NOT NULL,

)

CREATE TABLE ZAKAZI

(

Kod\_zakaza INT PRIMARY KEY,

Kod\_klienta INT FOREIGN KEY REFERENCES KLIENT\_1(Kod\_klienta),

Kod\_otelya INT FOREIGN KEY REFERENCES OTELI\_1(Kod\_otelya),

FIO VARCHAR(250) NOT NULL,

Strana INT FOREIGN KEY REFERENCES STRANI(Kod\_strani),

Kurort INT FOREIGN KEY REFERENCES KURORTI(Kod\_kurorta),

Otel VARCHAR(250) NOT NULL,

Kategoria\_otelya VARCHAR(250) NOT NULL,

Data\_otiezda DATE NOT NULL,

Prodolzhitelnost VARCHAR(250) NOT NULL,

)

GO

USE BD\_OTELI

INSERT KLIENT VALUES

('1','Иванов Иван Иванович','156012','741852963','ул. Мамедова, д.1, кв.6','89157446633','ivanov@mail.ru'),

('2','Петров Петр Петрович','741852','789456123','ул. Махмедова, д.2, кв.7','89190512020','petrov@mail.ru'),

('3','Федоров Федор Федорович','789456','321654987','ул. Самедова, д.3, кв.8','89156322014','fedorov@mail.ru')

INSERT KLIENT\_1 VALUES

('1','Егоров Евгений Иванович','741012','123852963','ул. Егорова, д.44, кв.21','89157206633','egorov@mail.ru'),

('2','Воронин Денис Петрович','987852','456456123','ул. Березова, д.55, кв.22','89150747020','voronin@mail.ru'),

('3','Рыбин Олег Федорович','632456','789654987','ул. Самарского, д.66, кв.23','89196322003','ribin@mail.ru')

INSERT KLIENT\_2 VALUES

('1','Васин Тимур Иванович','963012','741852963','ул. Фамедова, д.11, кв.66','89150006633','vasin@mail.ru'),

('2','Роваль Яков Петрович','654852','789456123','ул. Рахмедова, д.22, кв.77','89194442020','roval@mail.ru'),

('3','Пиронов Леонид Федорович','321456','321654987','ул. Бамедова, д.33, кв.88','89157772014','pironov@mail.ru')

INSERT STRANI VALUES

('1','Россия'),

('2','Германия'),

('3','Италия')

INSERT OTELI VALUES

('1','1','1','3 звезды','Олимп','Лучший отель в России','photo1.jpg'),

('2','2','2','4 звезды','Виктор','Лучший отель в Германии','photo2.jpg'),

('3','3','3','5 звезд','Дионис','Лучший отель в Италии','photo3.jpg')

INSERT OTELI\_1 VALUES

('1','1','1','3 звезды','Зевс','Лучший отель в Румынии','photo4.jpg'),

('2','2','2','4 звезды','Посейдон','Лучший отель в Греции','photo5.jpg'),

('3','3','3','5 звезд','Геркулес','Лучший отель в Болгарии','photo6.jpg')

INSERT OTELI\_2 VALUES

('1','1','1','3 звезды','Аполон','Лучший отель в Дании','photo7.jpg'),

('2','2','2','4 звезды','Борис','Лучший отель в Литве','photo8.jpg'),

('3','3','3','5 звезд','Яромир','Лучший отель в Латвии','photo9.jpg')

INSERT KURORTI VALUES

('1','1','Швеция','Шведский','Лучший курорт в Швеции','photo10.jpg'),

('2','2','Дания','Датский','Лучший курорт в Дании','photo11.jpg'),

('3','3','Франция','Французский','Лучший курорт во Франции','photo12.jpg')

INSERT ZAKAZI VALUES

('1','1','1','Денисов Денис Денисович','1','1','Отель Леонид','3 звезды','07.12.2023','1 неделя'),

('2','2','2','Олегов Олег Олегович','2','2','Отель Валериус','4 звезды','08.12.2023','2 недели'),

('3','3','3','Борисов Борис Борисович','3','3','Отель Василиус','5 звезд','09.12.2023','3 недели')

GO

**16. СОЗДАНИЕ ПРИЛОЖЕНИЯ**

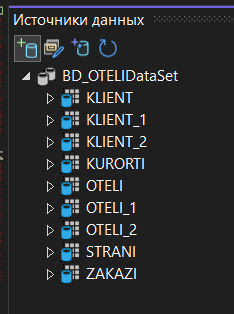


Рис. 14 Подключение БД



Рис. 15 Титульная форма

Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace BD\_OTELI

{

public partial class Titulnaya : Form

{

public Titulnaya()

{

InitializeComponent();

}

private void Fon\_Click(object sender, EventArgs e)

{

}

private void Avtorizacia\_knopka\_Click(object sender, EventArgs e)

{

Avtorizacia frm = new Avtorizacia();

frm.ShowDialog();

this.Hide();

}

private void Vihod\_iz\_progi\_Click(object sender, EventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Вы действительно хотите выйти из программы?", "Завершение программы",

MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (dialogResult == DialogResult.Yes)

{

Application.Exit();

}

else if (dialogResult == DialogResult.No)

{

//Nothing

}

}

}

}

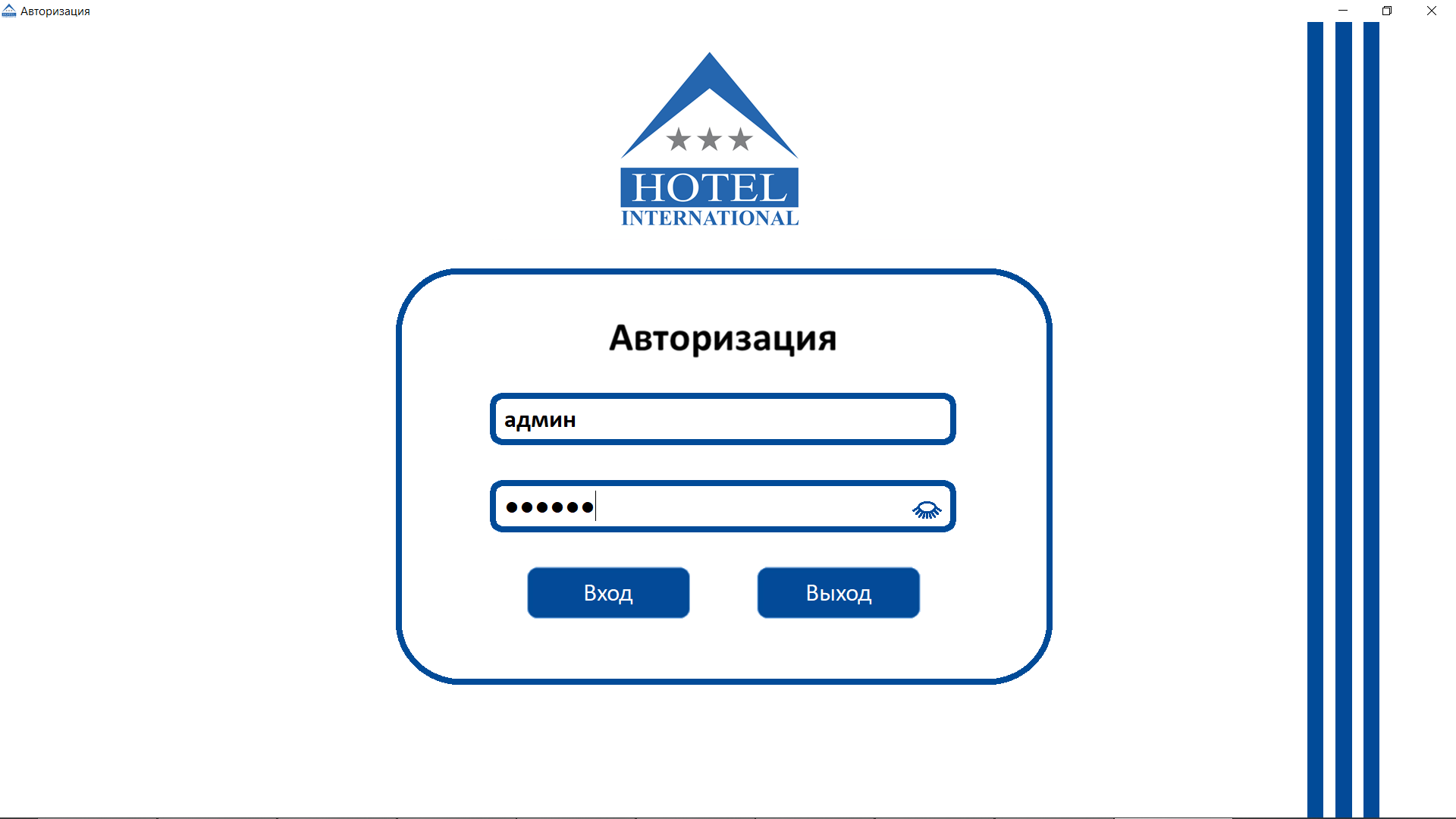


Рис. 16 Авторизация

Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace BD\_OTELI

{

public partial class Avtorizacia : Form

{

public Avtorizacia()

{

InitializeComponent();

}

private void Pole\_logina\_TextChanged(object sender, EventArgs e)

{

}

private void Glaz\_closed\_Click(object sender, EventArgs e)

{

Pole\_parolia.UseSystemPasswordChar = false;

Glaz\_closed.Visible = false;

Glaz\_open.Visible = true;

}

private void Glaz\_open\_Click(object sender, EventArgs e)

{

Pole\_parolia.UseSystemPasswordChar = true;

Glaz\_closed.Visible = true;

Glaz\_open.Visible = false;

}

private void Vhod\_knopka\_Click(object sender, EventArgs e)

{

if (Pole\_logina.Text == "юзер" & Pole\_parolia.Text == "123456")

{

Menu\_Usera frm = new Menu\_Usera();

frm.ShowDialog();

this.Hide();

}

else if (Pole\_logina.Text == "админ" & Pole\_parolia.Text == "741852")

{

Menu\_Admina frm = new Menu\_Admina();

frm.ShowDialog();

this.Hide();

}

else

{

MessageBox.Show("Неверный логин или пароль!", "Сообщение об ошибке", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void Vihod\_knopka\_Click(object sender, EventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Вы действительно хотите выйти из окна авторизации?", "Выход из авторизации",

MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialogResult == DialogResult.Yes)

{

Titulnaya frm = new Titulnaya();

frm.ShowDialog();

this.Hide();

}

else if (dialogResult == DialogResult.No)

{

//Nothing

}

}

}

}



Рис. 17 Меню Администратора

Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace BD\_OTELI

{

public partial class Menu\_Admina : Form

{

public Menu\_Admina()

{

InitializeComponent();

}

private void Vihod\_iz\_akka\_Click(object sender, EventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Вы действительно хотите выйти из аккаунта?", "Выход из аккаунта",

MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialogResult == DialogResult.Yes)

{

Avtorizacia frm = new Avtorizacia();

frm.ShowDialog();

this.Hide();

}

else if (dialogResult == DialogResult.No)

{

//Nothing

}

}

private void Bolee\_2\_Click(object sender, EventArgs e)

{

Bolee\_2 frm = new Bolee\_2();

frm.ShowDialog();

this.Hide();

}

private void Zakazi\_Click(object sender, EventArgs e)

{

Zakazi frm = new Zakazi();

frm.ShowDialog();

this.Hide();

}

private void Strani\_Click(object sender, EventArgs e)

{

Strani frm = new Strani();

frm.ShowDialog();

this.Hide();

}

}

}

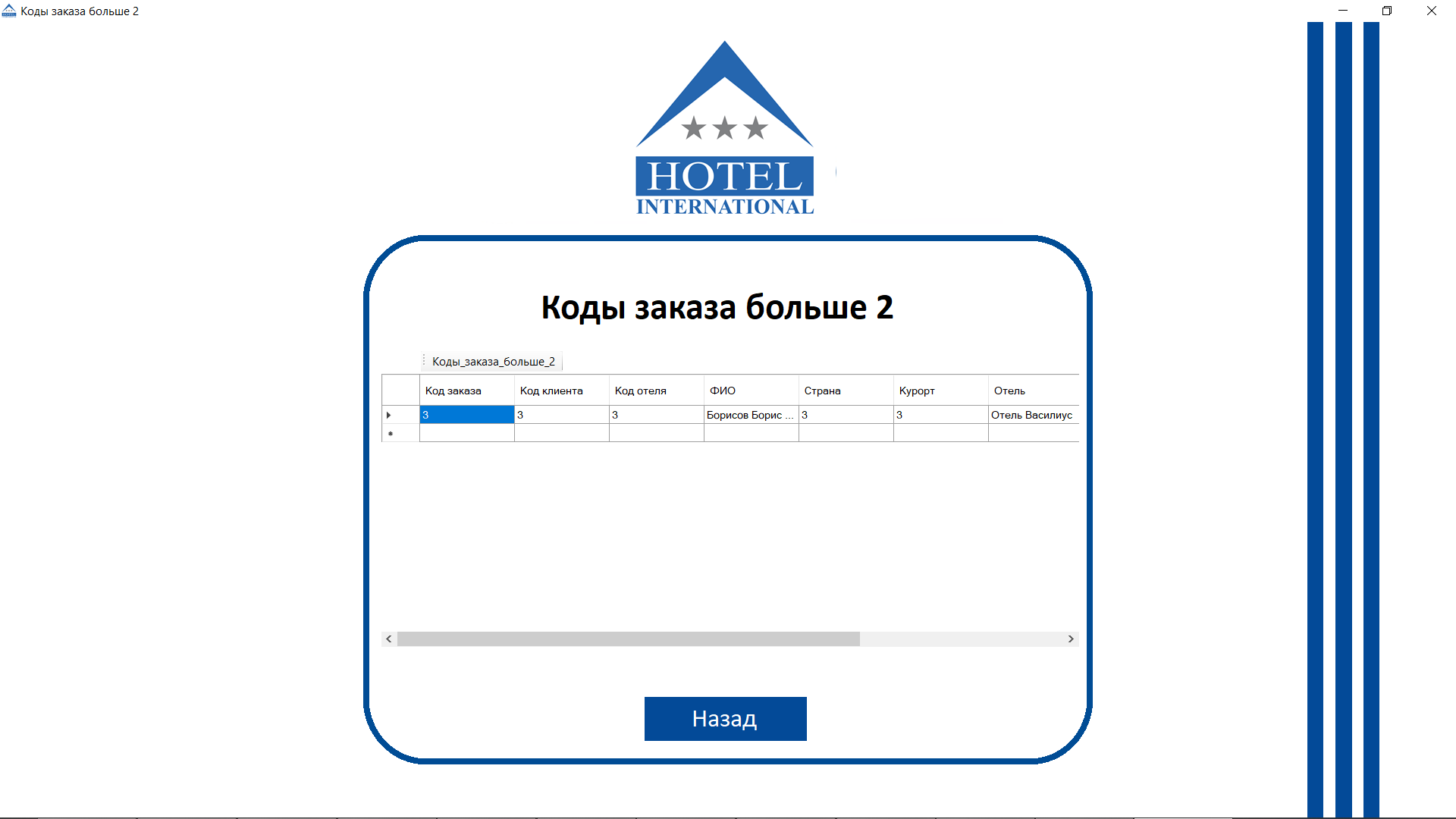


Рис. 18 Коды заказа больше 2

Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace BD\_OTELI

{

public partial class Bolee\_2 : Form

{

public Bolee\_2()

{

InitializeComponent();

}

private void Nazad\_knopka\_Click(object sender, EventArgs e)

{

Menu\_Admina frm = new Menu\_Admina();

frm.ShowDialog();

this.Hide();

}

private void Bolee\_2\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bD\_OTELIDataSet.ZAKAZI". При необходимости она может быть перемещена или удалена.

this.zAKAZITableAdapter.Fill(this.bD\_OTELIDataSet.ZAKAZI);

}

private void коды\_заказа\_больше\_2ToolStripButton\_Click(object sender, EventArgs e)

{

try

{

this.zAKAZITableAdapter.Коды\_заказа\_больше\_2(this.bD\_OTELIDataSet.ZAKAZI);

}

catch (System.Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

}

}

}

}

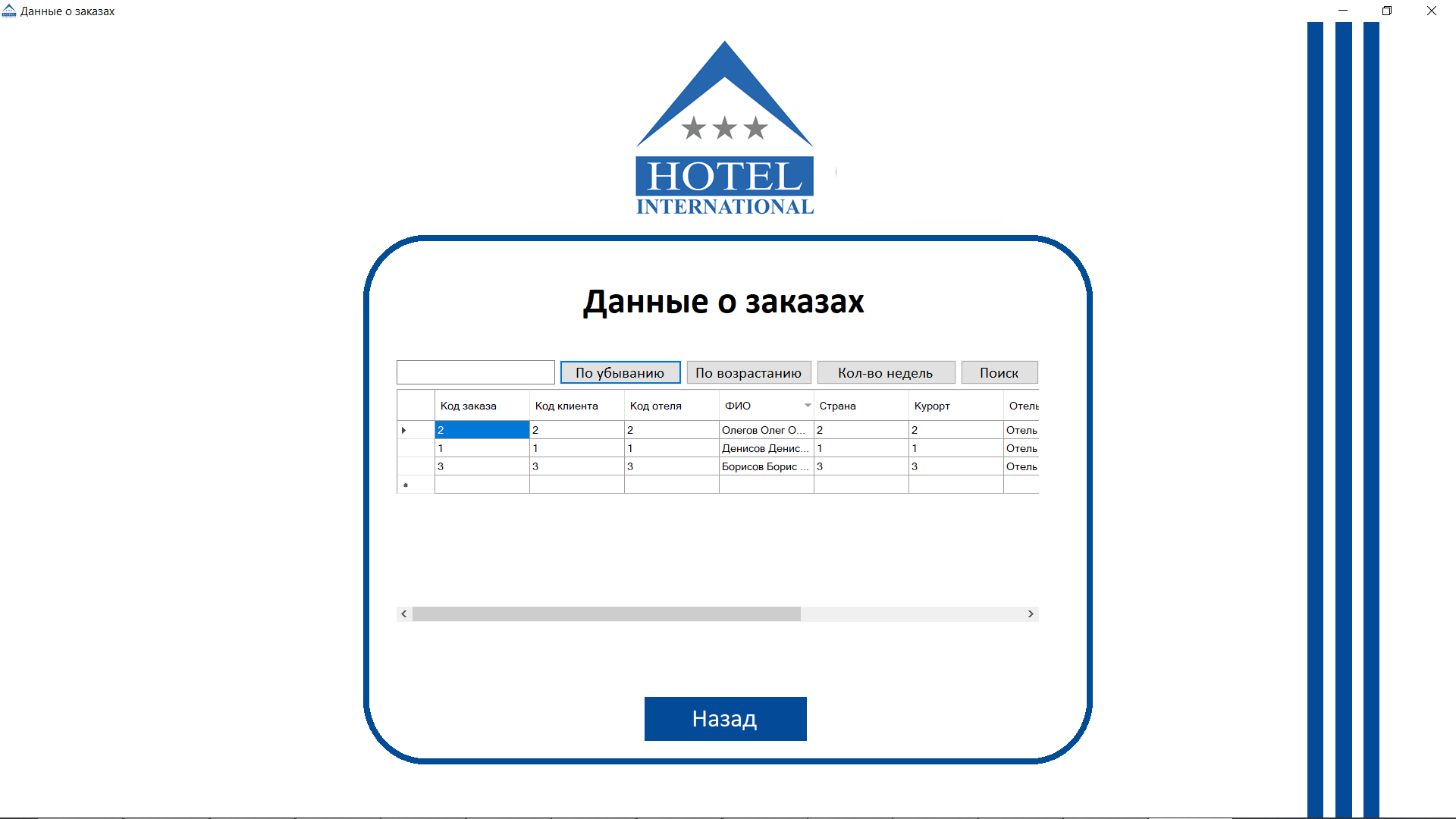


Рис. 19 Данные о заказах

Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace BD\_OTELI

{

public partial class Zakazi : Form

{

public Zakazi()

{

InitializeComponent();

}

private void Nazad\_knopka\_Click(object sender, EventArgs e)

{

Menu\_Admina frm = new Menu\_Admina();

frm.ShowDialog();

this.Hide();

}

private void Zakazi\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bD\_OTELIDataSet.ZAKAZI". При необходимости она может быть перемещена или удалена.

this.zAKAZITableAdapter.Fill(this.bD\_OTELIDataSet.ZAKAZI);

}

private void Knopka\_Po\_ubivaniy\_Click(object sender, EventArgs e)

{

zAKAZIBindingSource.Sort = "FIO Desc";

}

private void Knopka\_Po\_vozrastaniy\_Click(object sender, EventArgs e)

{

zAKAZIBindingSource.Sort = "FIO Asc";

}

private void Knopka\_Filtr\_\_Click(object sender, EventArgs e)

{

zAKAZIBindingSource.Filter = "[Prodolzhitelnost] LIKE'" + Pole\_vvoda\_.Text + "%'";

}

private void Knopka\_Poisk\_\_Click(object sender, EventArgs e)

{

{

for (int i = 0; i < dataGridView1.RowCount; i++)

{

dataGridView1.Rows[i].Selected = false;

for (int j = 0; j < dataGridView1.ColumnCount; j++)

if (dataGridView1.Rows[i].Cells[j].Value != null)

if (dataGridView1.Rows[i].Cells[j].Value.ToString().Contains(Pole\_vvoda\_.Text))

{

dataGridView1.Rows[i].Selected = true;

break;

}

}

}

}

}

}

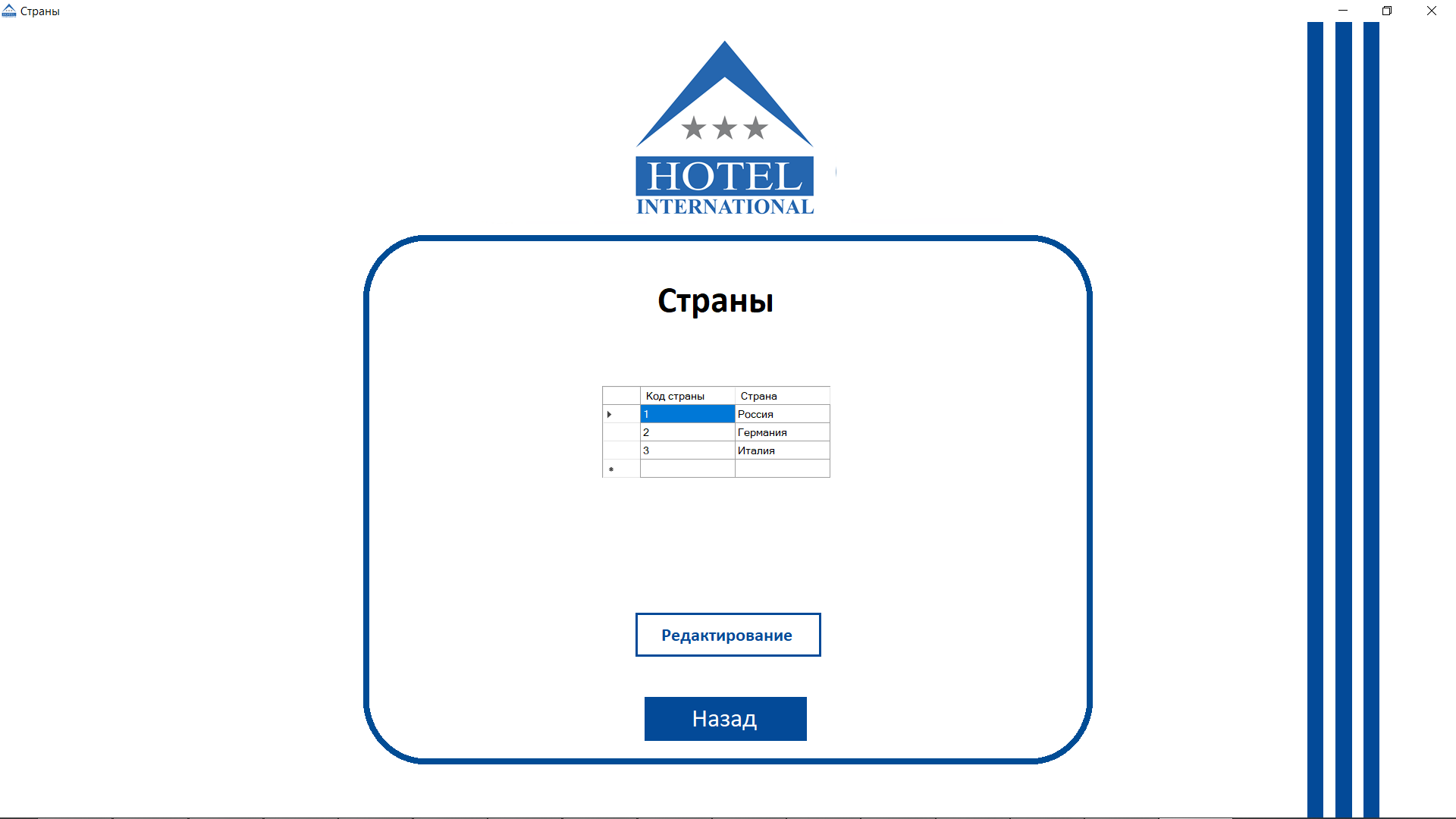


Рис. 20 Страны

Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace BD\_OTELI

{

public partial class Strani : Form

{

public Strani()

{

InitializeComponent();

}

private void Nazad\_knopka\_Click(object sender, EventArgs e)

{

Menu\_Admina frm = new Menu\_Admina();

frm.ShowDialog();

this.Hide();

}

private void Strani\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bD\_OTELIDataSet.STRANI". При необходимости она может быть перемещена или удалена.

this.sTRANITableAdapter.Fill(this.bD\_OTELIDataSet.STRANI);

}

private void Perehod\_Lenta\_Click(object sender, EventArgs e)

{

Lenta\_Strani frm = new Lenta\_Strani();

frm.ShowDialog();

this.Hide();

}

}

}

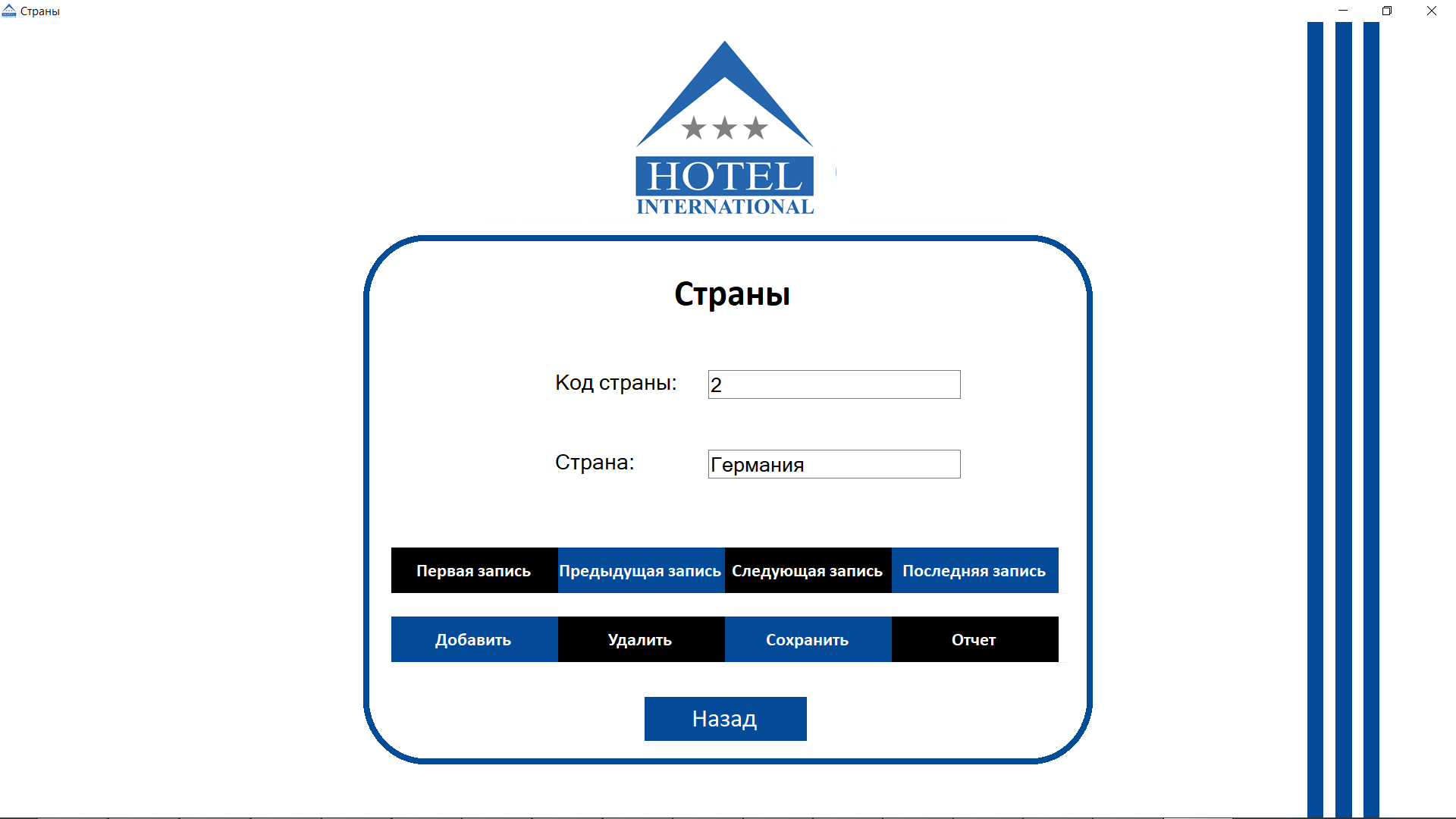


Рис. 21 Страны ленточная

Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace BD\_OTELI

{

public partial class Lenta\_Strani : Form

{

public Lenta\_Strani()

{

InitializeComponent();

}

private void sTRANIBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.sTRANIBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.bD\_OTELIDataSet);

}

private void Lenta\_Strani\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bD\_OTELIDataSet.STRANI". При необходимости она может быть перемещена или удалена.

this.sTRANITableAdapter.Fill(this.bD\_OTELIDataSet.STRANI);

}

private void Nazad\_knopka\_Click(object sender, EventArgs e)

{

Strani frm = new Strani();

frm.ShowDialog();

this.Hide();

}

private void Otchet\_Click(object sender, EventArgs e)

{

Ot\_Strani frm = new Ot\_Strani();

frm.ShowDialog();

this.Hide();

}

}

}

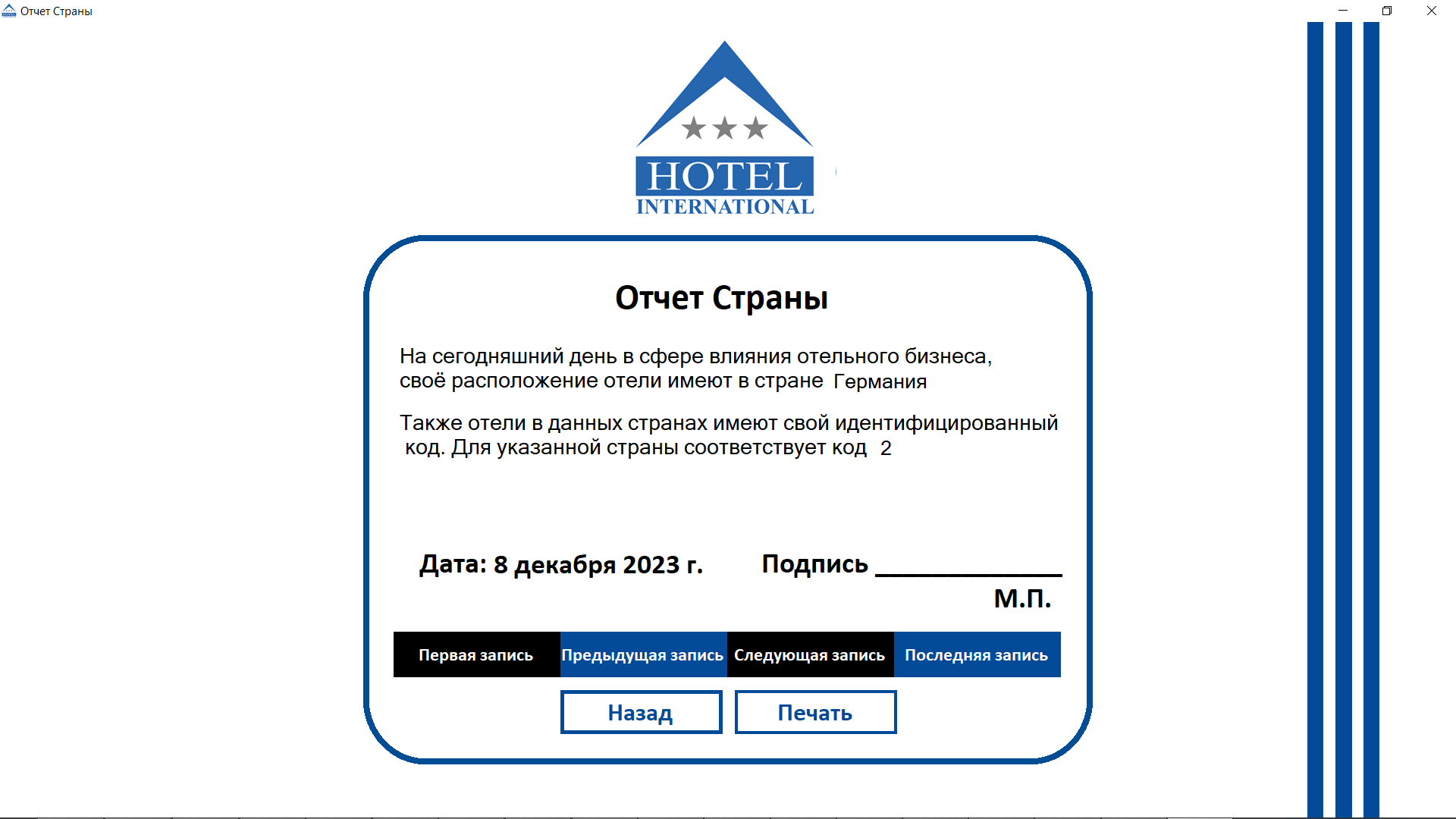


Рис. 22 Страны отчет

Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace BD\_OTELI

{

public partial class Ot\_Strani : Form

{

public Ot\_Strani()

{

InitializeComponent();

}

private void Ot\_Strani\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bD\_OTELIDataSet.STRANI". При необходимости она может быть перемещена или удалена.

this.sTRANITableAdapter.Fill(this.bD\_OTELIDataSet.STRANI);

}

private void Nazad\_Click(object sender, EventArgs e)

{

Lenta\_Strani frm = new Lenta\_Strani();

frm.ShowDialog();

this.Hide();

}

private void Pechatat\_Click(object sender, EventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Для печати необходимо подключить принтер!", "ВНИМАНИЕ!",

MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

private void richTextBox1\_TextChanged(object sender, EventArgs e)

{

}

private void Data\_Click(object sender, EventArgs e)

{

}

private void Podpis\_Click(object sender, EventArgs e)

{

}

private void timer1\_Tick(object sender, EventArgs e)

{

string datetext = DateTime.Now.ToLongDateString();

richTextBox1.Text = datetext;

}

}

}



Рис. 23 Меню Пользователя

Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace BD\_OTELI

{

public partial class Menu\_Usera : Form

{

public Menu\_Usera()

{

InitializeComponent();

}

private void Vihod\_iz\_akka\_Click(object sender, EventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Вы действительно хотите выйти из аккаунта?", "Выход из аккаунта",

MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialogResult == DialogResult.Yes)

{

Avtorizacia frm = new Avtorizacia();

frm.ShowDialog();

this.Hide();

}

else if (dialogResult == DialogResult.No)

{

//Nothing

}

}

private void Infa\_ob\_Oteli\_Click(object sender, EventArgs e)

{

Infa\_ob\_Oteli frm = new Infa\_ob\_Oteli();

frm.ShowDialog();

this.Hide();

}

private void Infa\_ob\_Oteli1\_Click(object sender, EventArgs e)

{

Infa\_ob\_Oteli1 frm = new Infa\_ob\_Oteli1();

frm.ShowDialog();

this.Hide();

}

private void Infa\_ob\_Oteli2\_Click(object sender, EventArgs e)

{

Infa\_ob\_Oteli2 frm = new Infa\_ob\_Oteli2();

frm.ShowDialog();

this.Hide();

}

}

}

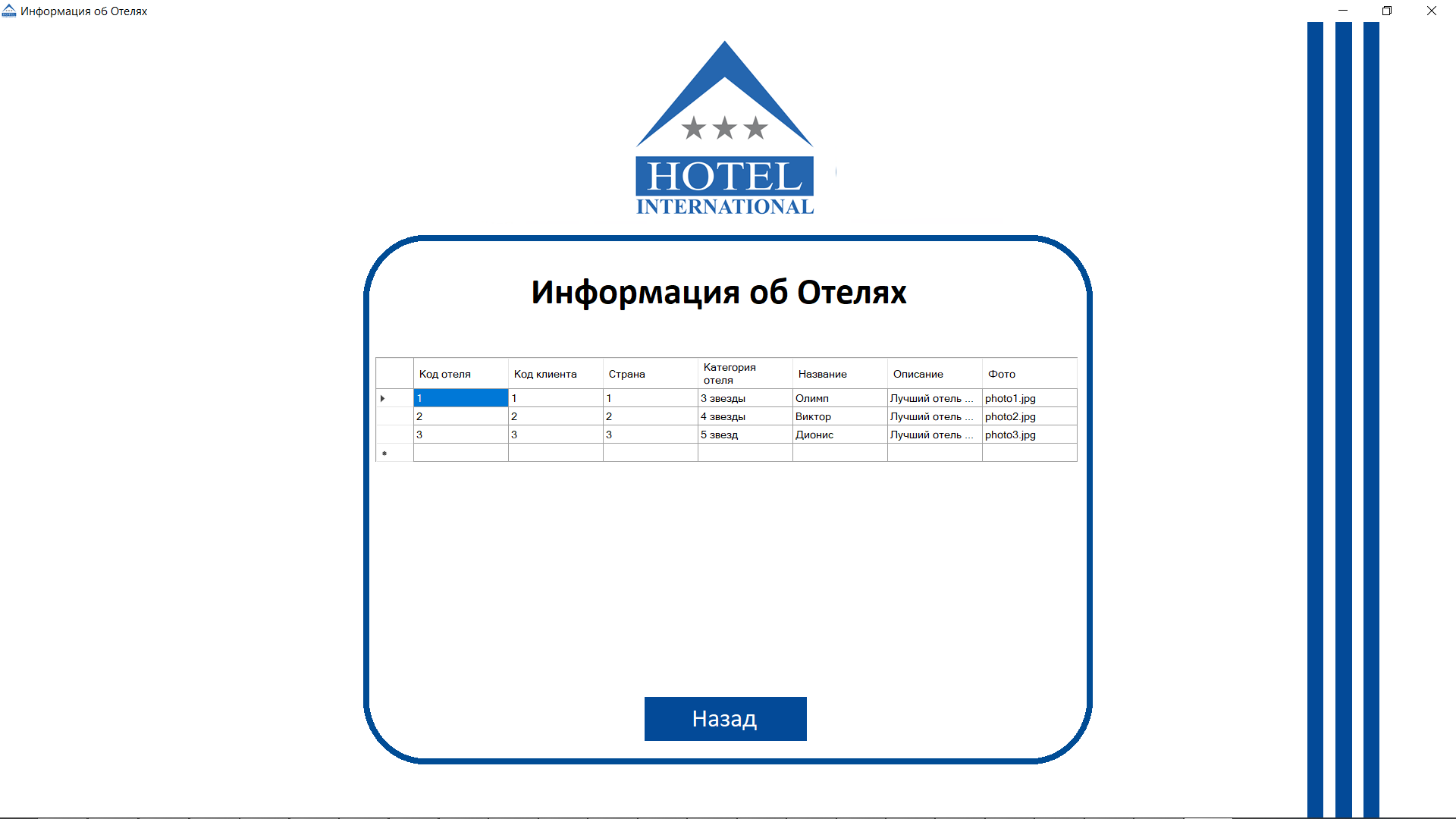


Рис. 24 Информация об Отелях

Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace BD\_OTELI

{

public partial class Infa\_ob\_Oteli : Form

{

public Infa\_ob\_Oteli()

{

InitializeComponent();

}

private void Nazad\_knopka\_Click(object sender, EventArgs e)

{

Menu\_Usera frm = new Menu\_Usera();

frm.ShowDialog();

this.Hide();

}

private void Infa\_ob\_Oteli\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bD\_OTELIDataSet.OTELI". При необходимости она может быть перемещена или удалена.

this.oTELITableAdapter.Fill(this.bD\_OTELIDataSet.OTELI);

}

}

}



Рис. 25 Информация об Отелях1

Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace BD\_OTELI

{

public partial class Infa\_ob\_Oteli1 : Form

{

public Infa\_ob\_Oteli1()

{

InitializeComponent();

}

private void Nazad\_knopka\_Click(object sender, EventArgs e)

{

Menu\_Usera frm = new Menu\_Usera();

frm.ShowDialog();

this.Hide();

}

private void Infa\_ob\_Oteli1\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bD\_OTELIDataSet.OTELI\_1". При необходимости она может быть перемещена или удалена.

this.oTELI\_1TableAdapter.Fill(this.bD\_OTELIDataSet.OTELI\_1);

}

}

}

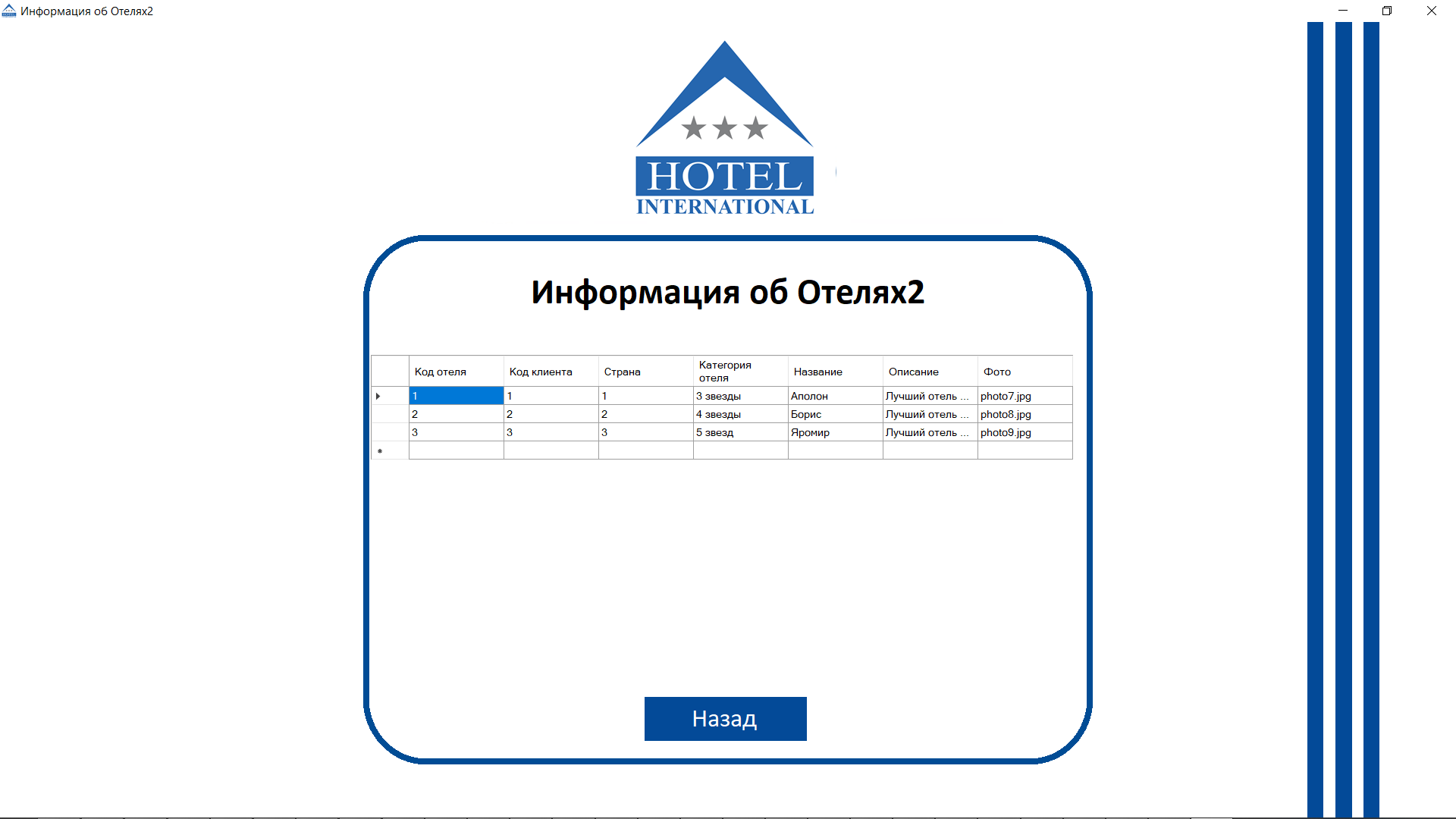


Рис. 26 Информация об Отелях2

Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace BD\_OTELI

{

public partial class Infa\_ob\_Oteli2 : Form

{

public Infa\_ob\_Oteli2()

{

InitializeComponent();

}

private void Nazad\_knopka\_Click(object sender, EventArgs e)

{

Menu\_Usera frm = new Menu\_Usera();

frm.ShowDialog();

this.Hide();

}

private void Infa\_ob\_Oteli2\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "bD\_OTELIDataSet.OTELI\_2". При необходимости она может быть перемещена или удалена.

this.oTELI\_2TableAdapter.Fill(this.bD\_OTELIDataSet.OTELI\_2);

}

}

}

**17. АЛГОРИТМ В ВИДЕ БЛОК-СХЕМЫ (ГОСТ 19.701)**

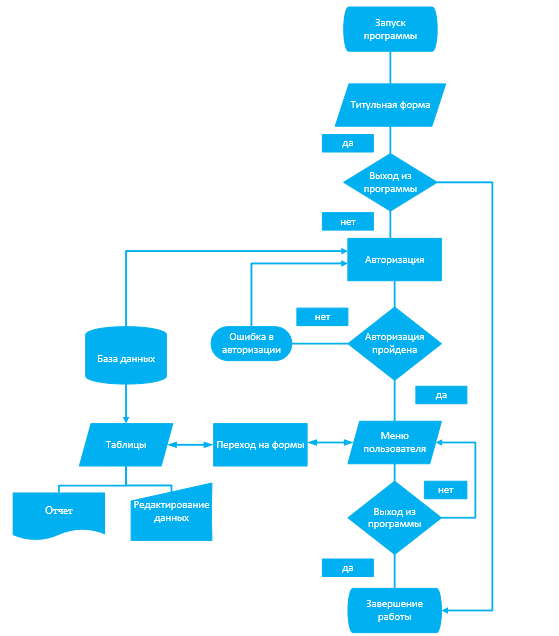


Рис. 27 Алгоритм в виде блок-схемы (ГОСТ 19.701)

**18. АЛГОРИТМ В ВИДЕ ТАБЛИЦ (ГОСТ 2.105)**

|  |  |  |
| --- | --- | --- |
| **Окна программы** | **Содержимое** | **Назначение** |
| **Титульная форма** | Заголовок формы | — |
| Кнопка «Вход» | Переход на форму «Авторизация». |
| Кнопка «Выход из приложения». | Завершение работы |
| **Авторизация** | Заголовок формы | — |
| Поля ввода логина и пароля | Для входа в аккаунт |
| Кнопка «Вход». | Переход на форму «Меню Пользователя». |
| Кнопка «Назад». | Переход на форму «Титульная форма». |
| **Меню Пользователя** | Заголовок формы. | — |
| Кнопка «Форма1». | Переход на форму «Форма1». |
| Кнопка «Форма2». | Переход на форму «Форма2». |
| Кнопка «Форма3». | Переход на форму «Форма3». |
| Кнопка «Выход». | Переход на форму «Авторизация». |
| **Меню Администратора** | Заголовок формы. | — |
| Кнопка «Форма1». | Переход на форму «Форма1». |
| Кнопка «Форма2». | Переход на форму «Форма2». |
| Кнопка «Форма3». | Переход на форму «Форма3». |
| Кнопка «Выход». | Переход на форму «Авторизация». |

**19. АЛГОРИТМ В ВИДЕ ТЕКСТОВОГО ОПИСАНИЯ (ГОСТ 24.301)**

Программа представляет собой многофункциональную платформу. Она включает в себя несколько ключевых компонентов, которые делают её незаменимым помощником для пользователя и администратора.

**ТИТУЛЬНАЯ ФОРМА**

Это первая страница, которую видит пользователь при открытии приложения. Оно содержит заголовок, кнопку для входа в авторизацию и кнопку выхода из программы.

**АВТОРИЗАЦИЯ**

Здесь пользователь вводит свои данные для входа в программу. Программа поддерживает однопользовательский режим.

**МЕНЮ ПОЛЬЗОВАТЕЛЯ**

После успешной авторизации пользователь попадает в меню пользователя, где он может нажимать на кнопки для перехода на различные формы.

**МЕНЮ АДМИНИСТРАТОРА**

Администратор может нажимать на кнопки для перехода на различные формы программы.

**ОКНО РЕДАКТИРОВАНИЯ**

Это окно используется только администратором для редактирования информации в таблицах БД. Оно включает в себя интуитивно понятный интерфейс для добавления, удаления, сохранения и изменения, передвижения по записям из БД.

**20. ПРОТОТИП ПРОГРАММЫ**

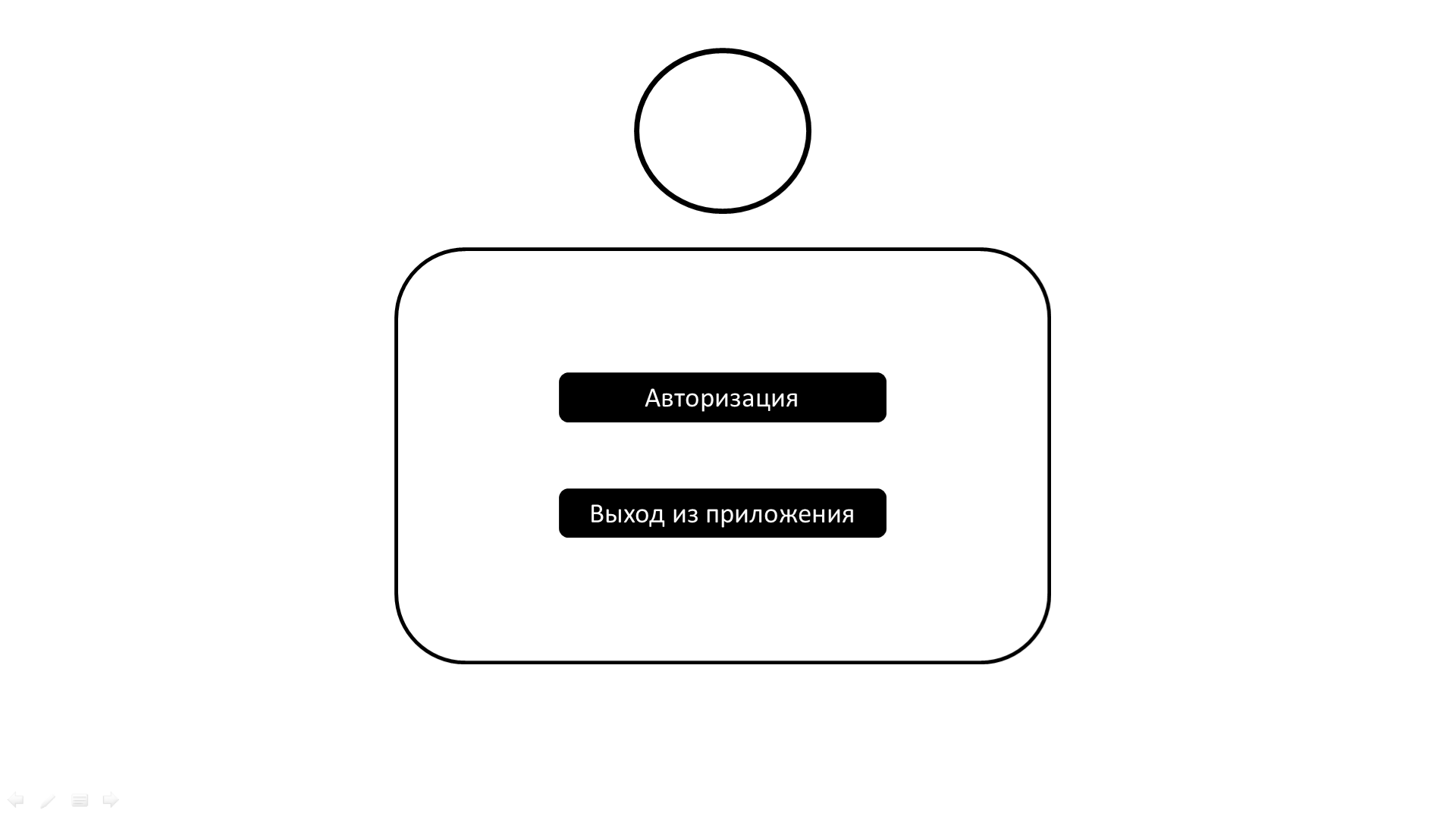


Рис. 28 Прототип

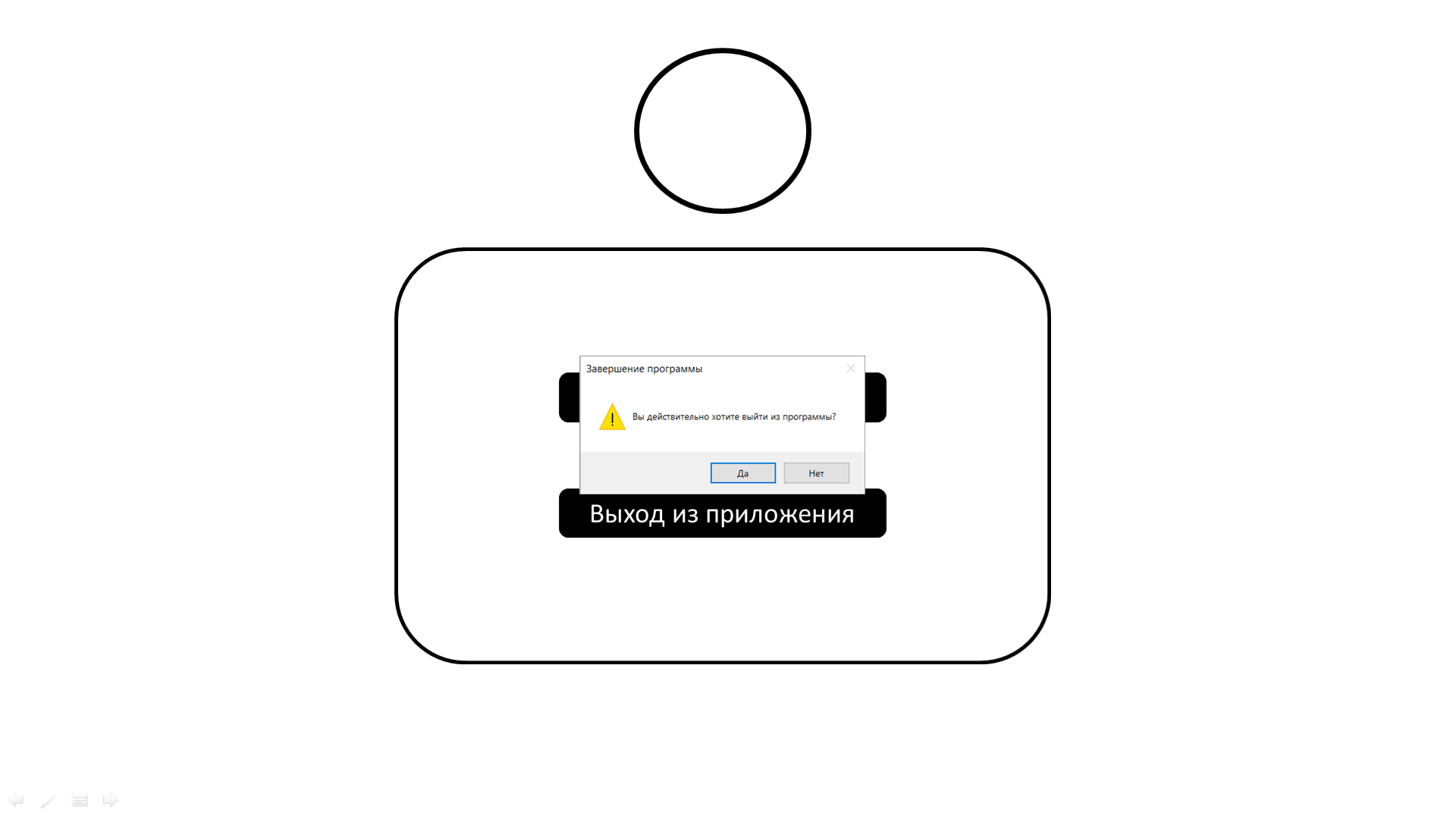


Рис. 29 Прототип

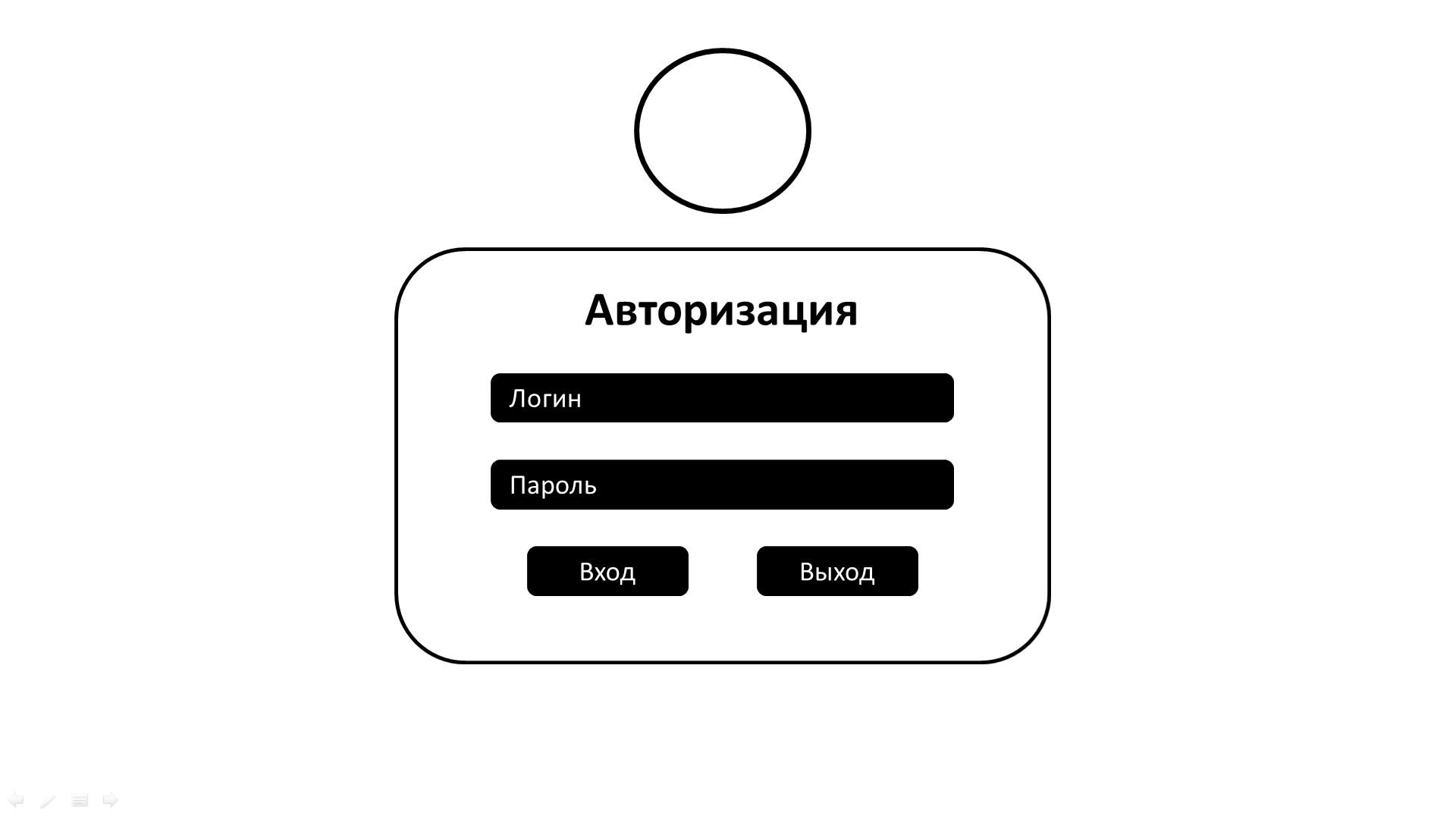


Рис. 30 Прототип



Рис. 31 Прототип

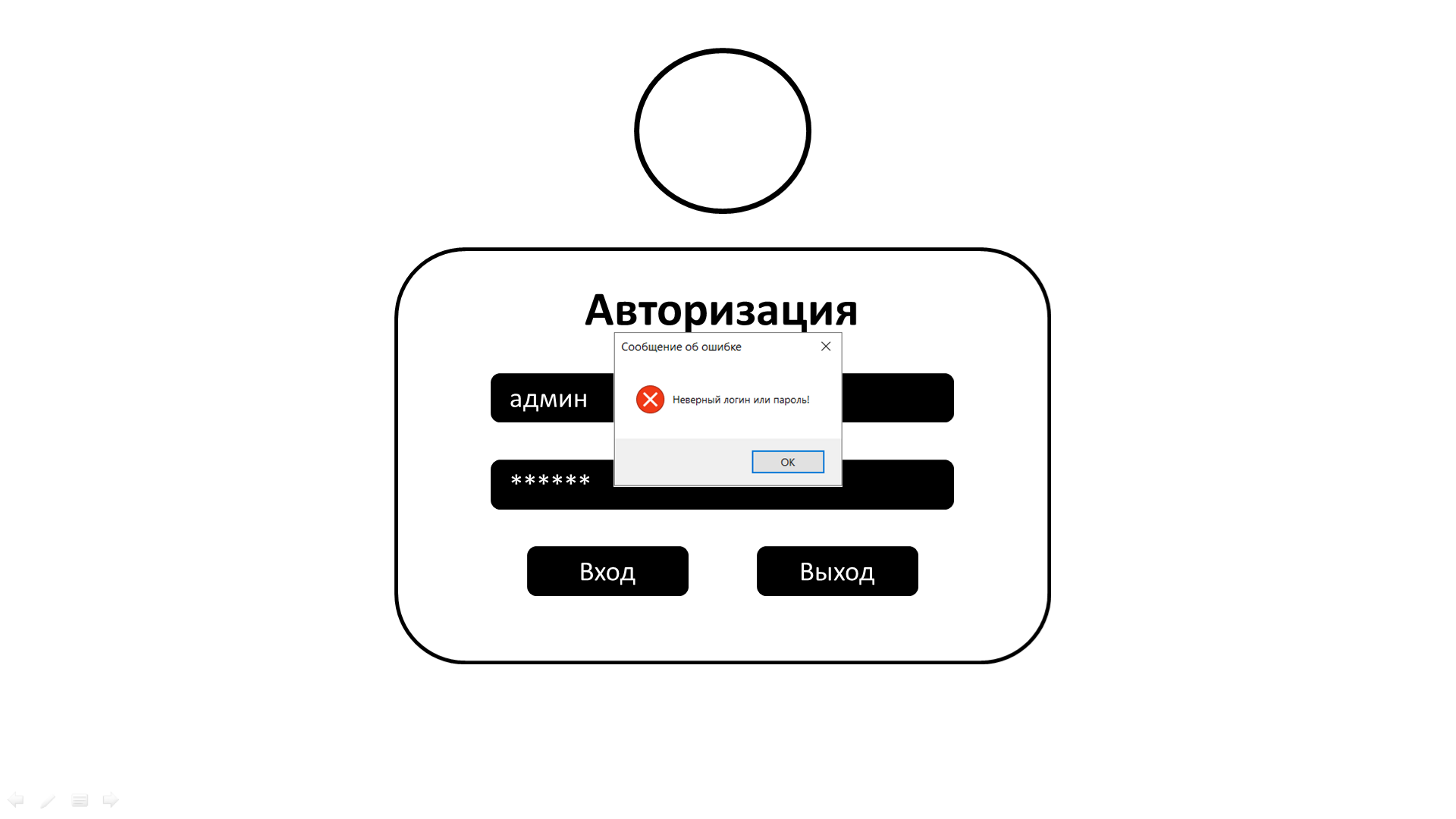


Рис. 32 Прототип

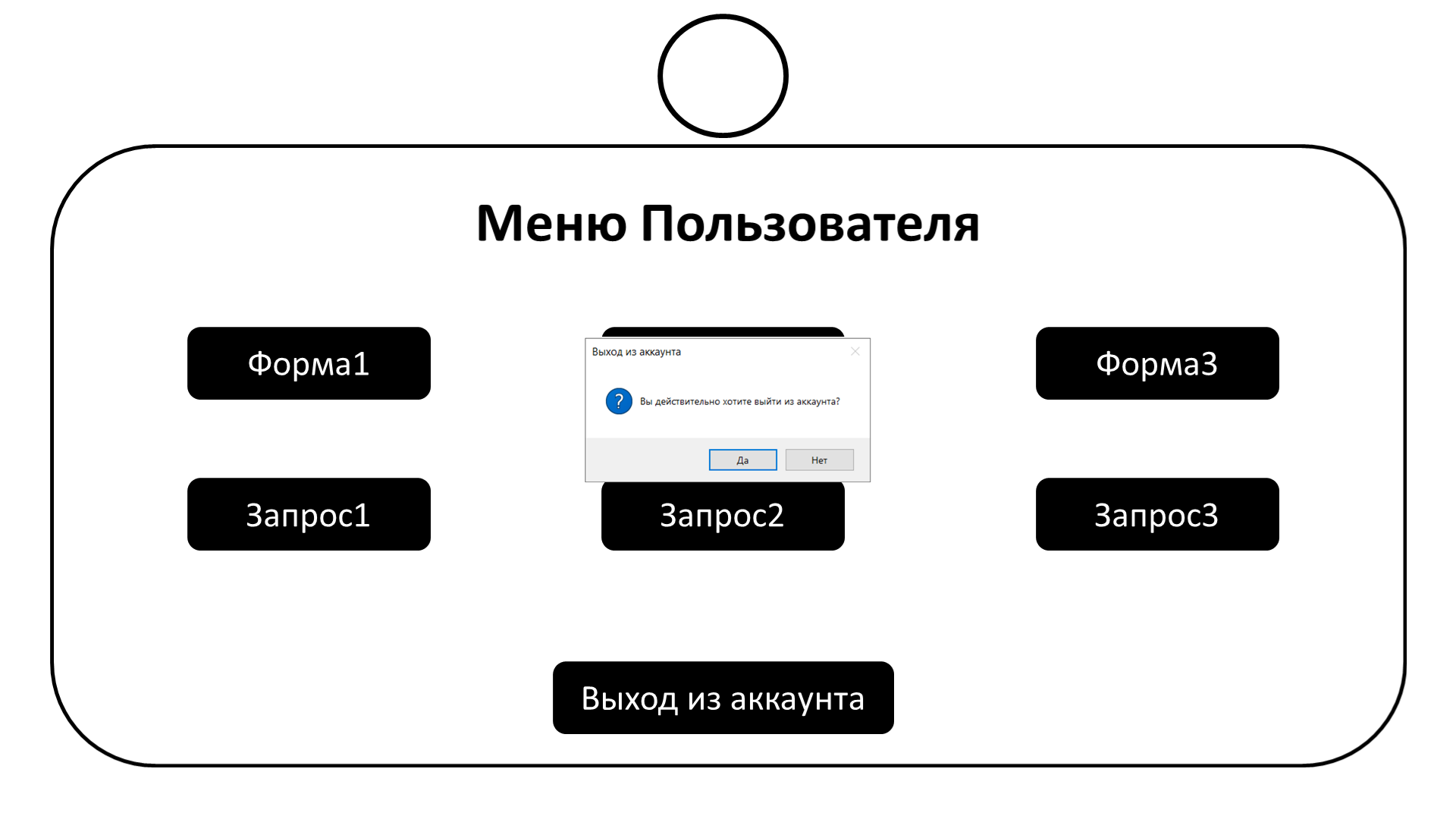


Рис. 33 Прототип

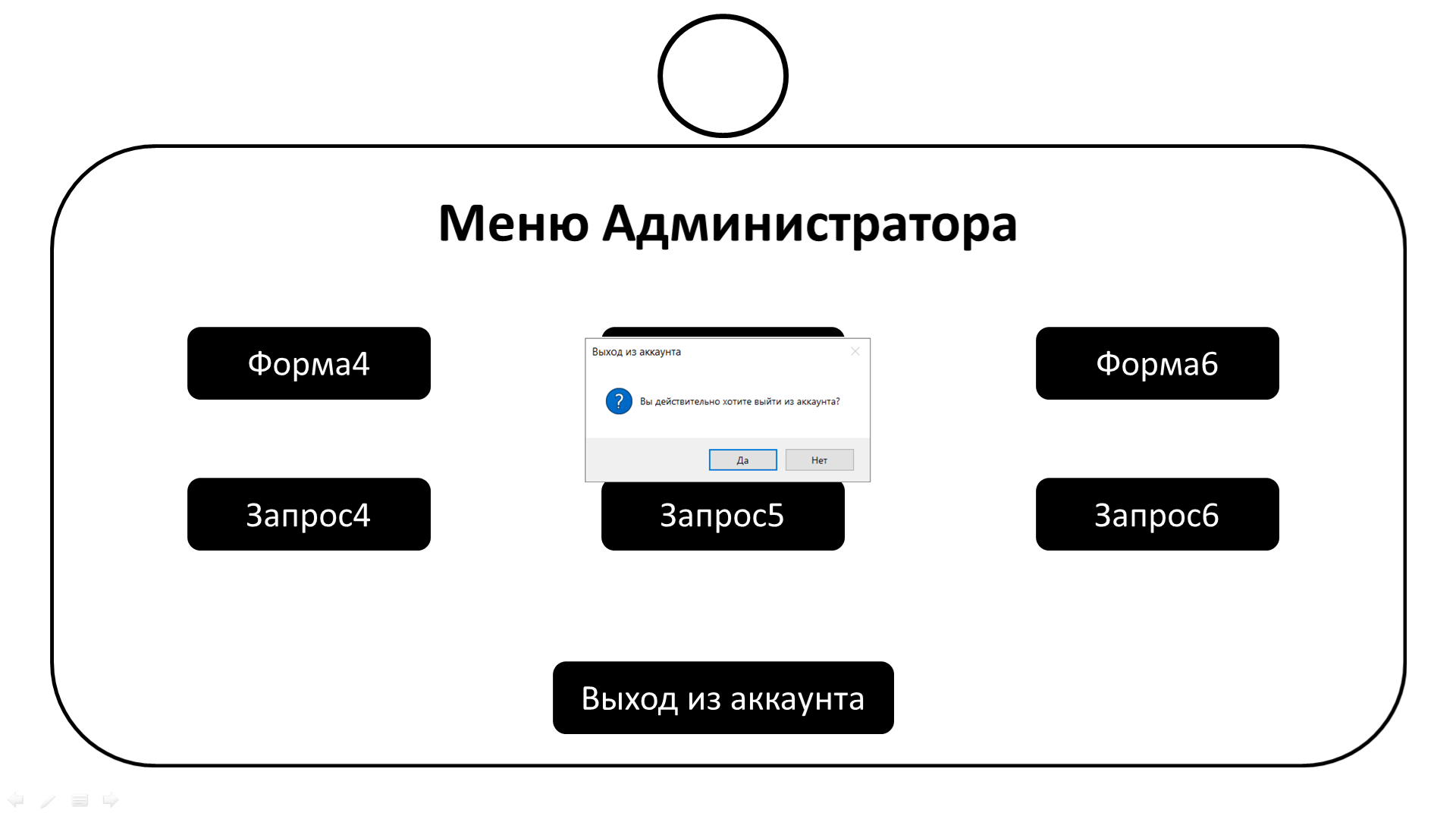


Рис. 34 Прототип

**21. ER-ДИАГРАММА (ПРОКАТ АВТОМОБИЛЕЙ)**

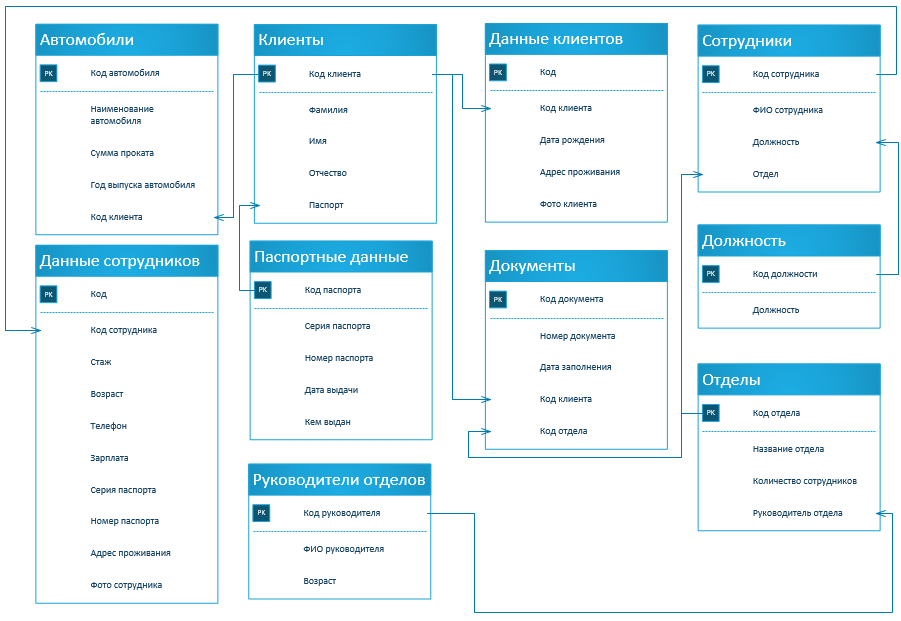


Рис. 35 ER-диаграмма

**22. АЛГОРИТМ В ВИДЕ БЛОК-СХЕМЫ (ПРОКАТ АВТОМОБИЛЕЙ)**

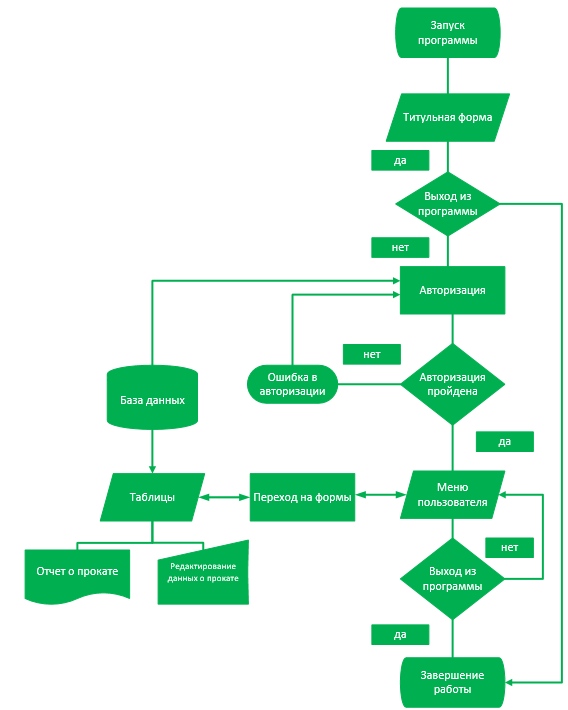


Рис. 36 Блок-схема

**23. АЛГОРИТМ В ВИДЕ ТАБЛИЦ (ПРОКАТ АВТОМОБИЛЕЙ)**

|  |  |  |
| --- | --- | --- |
| **Окна программы** | **Содержимое** | **Назначение** |
| **Титульная форма** | Заголовок формы | — |
| Кнопка «Вход» | Переход на форму «Авторизация». |
| Кнопка «Выход из приложения». | Завершение работы |
| **Авторизация** | Заголовок формы | — |
| Поля ввода логина и пароля | Для входа в аккаунт |
| Кнопка «Вход». | Переход на форму «Меню Пользователя». |
| Кнопка «Назад». | Переход на форму «Титульная форма». |
| **Меню Пользователя** | Заголовок формы. | — |
| Кнопка «Автомобили». | Переход на форму «Автомобили». |
| Кнопка «Цена за прокат». | Переход на форму «Цена за прокат». |
| Кнопка «Данные о прокате». | Переход на форму «Данные о прокате». |
| Кнопка «Выход». | Переход на форму «Авторизация». |
| **Меню Администратора** | Заголовок формы. | — |
| Кнопка «Информация о клиентах». | Переход на форму «Информация о клиентах». |
| Кнопка «Прокаты клиентов». | Переход на форму «Прокаты клиентов». |
| Кнопка «Информация о сотрудниках». | Переход на форму «Информация о сотрудниках». |
| Кнопка «Выход». | Переход на форму «Авторизация». |

**24. АЛГОРИТМ В ВИДЕ ТЕКСТОВОГО ОПИСАНИЯ (ПРОКАТ АВТОМОБИЛЕЙ)**

Программа представляет собой многофункциональную платформу. Она включает в себя несколько ключевых компонентов, которые делают её незаменимым помощником для пользователя и администратора.

**ТИТУЛЬНАЯ ФОРМА**

Это первая страница, которую видит пользователь при открытии приложения. Оно содержит заголовок, кнопку для входа в авторизацию и кнопку выхода из программы.

**АВТОРИЗАЦИЯ**

Здесь пользователь вводит свои данные для входа в программу. Программа поддерживает однопользовательский режим.

**МЕНЮ ПОЛЬЗОВАТЕЛЯ**

После успешной авторизации пользователь попадает в меню пользователя, где он может нажимать на кнопки для перехода на различные формы. На формах содержится информация о возможности проката автомобилей и цены за прокат.

**МЕНЮ АДМИНИСТРАТОРА**

Администратор может нажимать на кнопки для перехода на различные формы программы. Ему предоставлена информация о клиентах, сотрудниках и данных о прокате автомобилей.

**ОКНО РЕДАКТИРОВАНИЯ**

Это окно используется только администратором для редактирования информации в таблицах БД. Оно включает в себя интуитивно понятный интерфейс для добавления, удаления, сохранения и изменения, передвижения по записям из БД. Администратор может работать с данными о клиентах, сотрудниках и автомобилях.

**25. БАЗА ДАННЫХ (ПРОКАТ АВТОМОБИЛЕЙ)**

Код:

CREATE DATABASE BD\_PROKAT\_AVTO

GO

USE BD\_PROKAT\_AVTO

CREATE TABLE PASPORTNIE\_DANNIE

(

Kod\_pasporta INT PRIMARY KEY,

Seria\_pasporta INT NOT NULL,

Nomer\_pasporta INT NOT NULL,

Data\_vidachi DATE NOT NULL,

Kem\_vidan VARCHAR(250) NOT NULL,

)

CREATE TABLE KLIENTI

(

Kod\_klienta INT PRIMARY KEY,

Familia VARCHAR(250) NOT NULL,

Imya VARCHAR(250) NOT NULL,

Otshestvo VARCHAR(250) NOT NULL,

Pasport INT FOREIGN KEY REFERENCES PASPORTNIE\_DANNIE(Kod\_pasporta),

)

CREATE TABLE DANNIE\_KLIENTA

(

Kod INT PRIMARY KEY,

Klient INT FOREIGN KEY REFERENCES KLIENTI(Kod\_klienta),

Data\_rozhdeniya DATE NOT NULL,

Adres\_prozhivaniya VARCHAR(250) NOT NULL,

Foto\_klienta VARCHAR(250) NOT NULL,

)

CREATE TABLE AVTOMOBILI

(

Kod\_avto INT PRIMARY KEY,

Naimenovanie\_avto VARCHAR(250) NOT NULL,

Summa\_prokata INT NOT NULL,

God\_vipuska\_avto INT NOT NULL,

Klient INT FOREIGN KEY REFERENCES KLIENTI(Kod\_klienta),

)

CREATE TABLE DOLZHNOST

(

Kod\_dolzhnosti INT PRIMARY KEY,

Dolzhnost VARCHAR(250) NOT NULL,

)

CREATE TABLE RUKOVODITELI\_OTDELOV

(

Kod\_rukovoditeliya INT PRIMARY KEY,

FIO\_rukovoditeliya VARCHAR(250) NOT NULL,

Vozrast INT NOT NULL,

)

CREATE TABLE OTDELI

(

Kod\_otdela INT PRIMARY KEY,

Nazvanie\_otdela VARCHAR(250) NOT NULL,

Kolichestvo\_sotrudnikov INT NOT NULL,

Rukovoditel\_otdela INT FOREIGN KEY REFERENCES RUKOVODITELI\_OTDELOV(Kod\_rukovoditeliya),

)

CREATE TABLE SOTRUDNIKI

(

Kod\_sotrudnika INT PRIMARY KEY,

FIO\_sotrudnika VARCHAR(250) NOT NULL,

Dolzhnost INT FOREIGN KEY REFERENCES DOLZHNOST(Kod\_dolzhnosti),

Otdel INT FOREIGN KEY REFERENCES OTDELI(Kod\_otdela),

)

CREATE TABLE DOKUMENTI

(

Kod\_dokumenta INT PRIMARY KEY,

Nomer\_dokumenta INT NOT NULL,

Data\_zapolneniya DATE NOT NULL,

Klient INT FOREIGN KEY REFERENCES KLIENTI(Kod\_klienta),

Otdel INT FOREIGN KEY REFERENCES OTDELI(Kod\_otdela),

)

CREATE TABLE DANNIE\_SOTRUDNIKOV

(

Kod INT PRIMARY KEY,

Sotrudnik INT FOREIGN KEY REFERENCES SOTRUDNIKI(Kod\_sotrudnika),

Stazh INT NOT NULL,

Vozrast INT NOT NULL,

Telefon VARCHAR(250) NOT NULL,

Zarplata INT NOT NULL,

Seria\_pasporta INT NOT NULL,

Nomer\_pasporta INT NOT NULL,

Adres\_prozhivaniya VARCHAR(250) NOT NULL,

Foto\_sotrudnika VARCHAR(250) NOT NULL,

)

CREATE TABLE DANNIE\_DLYA\_VHODA

(

Kod INT PRIMARY KEY,

Login VARCHAR(250) NOT NULL,

Parol VARCHAR(250) NOT NULL,

)

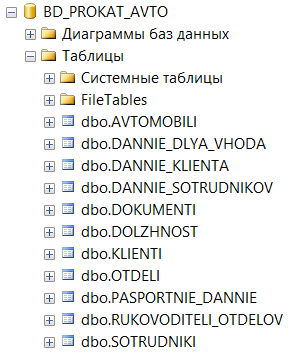


Рис. 37 Результат создания БД

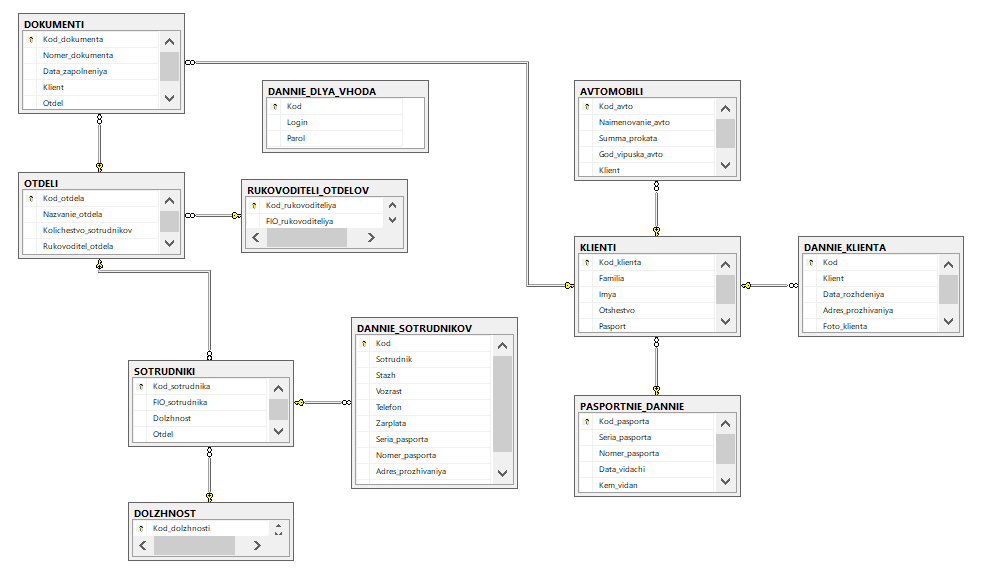


Рис. 38 Схема данных БД

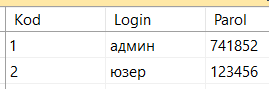


Рис. 39 Таблица «DANNIE\_DLYA\_VHODA»

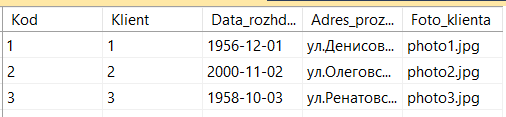


Рис. 40 Таблица «DANNIE\_KLIENTA»

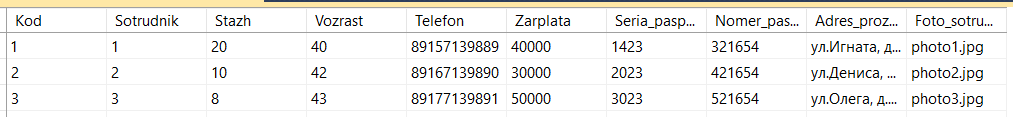


Рис. 41 Таблица «DANNIE SOTRUDNIKOV»

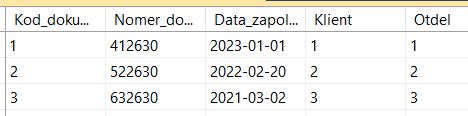


Рис. 42 Таблица «DOKUMENTI»

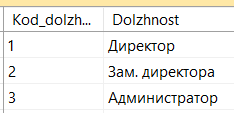


Рис. 43 Таблица «DOLZHNOST»

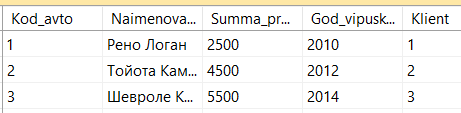


Рис. 44 Таблица «AVTOMOBILI»

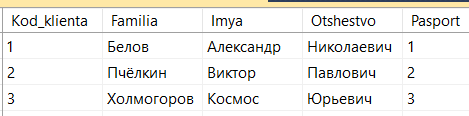


Рис. 45 Таблица «KLIENTI»

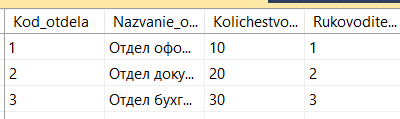


Рис. 46 Таблица «OTDELI»

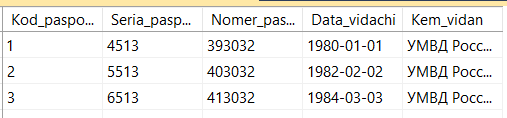


Рис. 47 Таблица «PASPORTNIE\_DANNIE»

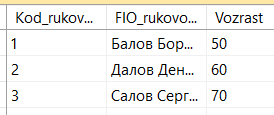


Рис. 48 Таблица «RUKOVODITELI\_OTDELOV»

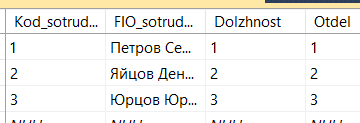


Рис. 49 Таблица «SOTRUDNIKI»

**26. ФОРМА АВТОРИЗАЦИИ (ПРОКАТ АВТОМОБИЛЕЙ)**

Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Drawing.Imaging;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Net.Mime.MediaTypeNames;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace BD\_PROKAT\_AVTO

{

public partial class Avtorizacia : Form

{

public Avtorizacia()

{

InitializeComponent();

}

private void Vhod\_knopka\_Click(object sender, EventArgs e)

{

if (Pole\_logina.Text == "юзер" & Pole\_parolia.Text == "123456")

{

Menu\_Usera frm = new Menu\_Usera();

frm.ShowDialog();

this.Hide();

}

else if (Pole\_logina.Text == "админ" & Pole\_parolia.Text == "741852")

{

Menu\_Admina frm = new Menu\_Admina();

frm.ShowDialog();

this.Hide();

}

else

{

MessageBox.Show("Неверный логин или пароль!", "Сообщение об ошибке", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void Vihod\_knopka\_Click(object sender, EventArgs e)

{

DialogResult dialogResult = MessageBox.Show("Вы действительно хотите выйти из окна авторизации?", "Выход из авторизации",

MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialogResult == DialogResult.Yes)

{

Titulnaya\_forma frm = new Titulnaya\_forma();

frm.ShowDialog();

this.Hide();

}

else if (dialogResult == DialogResult.No)

{

//Nothing

}

}

private void Glaz\_closed\_Click(object sender, EventArgs e)

{

Pole\_parolia.UseSystemPasswordChar = false;

Glaz\_closed.Visible = false;

Glaz\_open.Visible = true;

}

private void Glaz\_open\_Click(object sender, EventArgs e)

{

Pole\_parolia.UseSystemPasswordChar = true;

Glaz\_closed.Visible = true;

Glaz\_open.Visible = false;

}

private void Avtorizacia\_Load(object sender, EventArgs e)

{

}

}

}

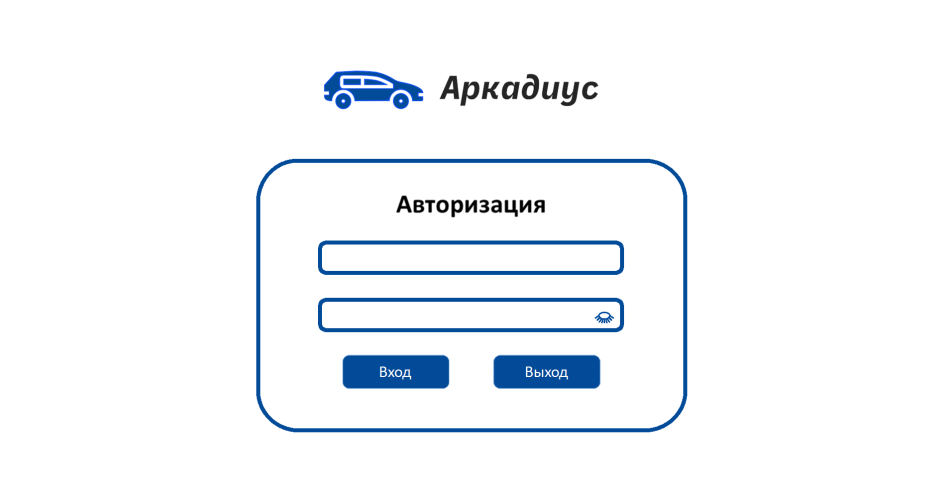


Рис. 50 Форма «Авторизация»

**27. РУКОВОДСТВО СИСТЕМНОМУ ПРОГРАММИСТУ (ГОСТ 19.503-79)**

**Программа «Прокат автомобилей»**

**Руководство системному программисту**

**Листов «5»**

г. Вышний Волочек

2024 г.

**СОДЕРЖАНИЕ**

1. ВВЕДЕНИЕ………………………………………………………………..…..74

2. СОДЕРЖАНИЕ РАЗДЕЛОВ ………………………………………...……....74

2.1. НАЗНАЧЕНИЕ ПРОГРАММЫ…………………………………………....74

2.2. УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ……………………………..74

2.3 ВЫПОЛНЕНИЕ ПРОГРАММЫ……………………………………………75

2.4. СООБЩЕНИЯ ОПЕРАТОРУ………………………………………………76

**1. ВВЕДЕНИЕ**

Информационная система «Прокат автомобиле й» предназначена для автоматизации процессов аренды автомобилей, повышения уровня сервиса и удобства пользователей. Она позволяет клиентам быстро и удобно забронировать автомобиль на определенное время, а также отслеживать статус своих заказов в режиме реального времени. Система также упрощает административные задачи, такие как учет автомобилей, контроль за состоянием машин и управление персоналом. Благодаря интеграции с другими системами, клиенты могут легко управлять своими бронированиями и просматривать доступные автомобили, а менеджеры проката получают возможность анализировать данные и оптимизировать рабочие процессы.

**2. СОДЕРЖАНИЕ РАЗДЕЛОВ**

**2.1. НАЗНАЧЕНИЕ ПРОГРАММЫ**

Назначение программы для информационной системы «Прокат автомобиле й» — это облегчение работы сотрудников и улучшение качества обслуживания клиентов. Программа позволяет автоматизировать многие рутинные операции, такие как ведение базы данных автомобилей, учёт свободных и занятых машин, бронирование и отслеживание статуса заказов. Кроме того, система предоставляет возможность клиентам самостоятельно управлять своими бронированиями, что значительно экономит время и упрощает процесс аренды автомобиля. Также программа помогает менеджерам проката контролировать работу персонала и анализировать эффективность работы проката в целом.

**2.2. УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ**

Для выполнения программы информационной системы «Прокат автомобиле й» необходимо наличие компьютера или другого устройства с доступом в интернет и установленной программой веб-браузера. Также необходимо зарегистрироваться в системе, создав учетную запись пользователя. После регистрации пользователю становится доступен весь функционал системы, включая возможность бронирования автомобилей, отслеживания статуса заказов, управления своими бронированиями и просмотра доступных автомобилей. Менеджеры проката также получают возможность анализа данных и оптимизации рабочих процессов.

Выполнение программы может быть затруднено при отсутствии доступа в интернет или при нестабильной работе программного обеспечения. В этом случае следует обратиться к администратору для устранения проблемы.

Требования к программным и аппаратным средствам:

* Процессор Pentium 133 или выше;
* Оперативная память 16 Мбайт или выше;
* Пространство на жестком диске для установки 45 Мбайт;
* Пространство для временных файлов 25 Мбайт;
* Наличие CD-ROM или дисковода, а также есть возможность установки всех компонентов через локальную или глобальную сеть;
* Монитор SVGA видео карта с буфером памяти Мбайт.

**2.3 ВЫПОЛНЕНИЕ ПРОГРАММЫ**

Информационная система «Прокат автомобиле й» выполняет ряд задач, направленных на оптимизацию работы проката и улучшение качества обслуживания клиентов. К таким задачам относятся автоматизация процессов аренды автомобилей, учет автомобилей и контроль за их состоянием, управление персоналом и анализ данных. Благодаря интеграции с другими системами, пользователи могут легко управлять своими бронированиями и отслеживать статус заказов, а менеджеры получают возможность оптимизации рабочих процессов и анализа эффективности работы проката. Основными функциями системы являются:

* Бронирование автомобилей
* Управление бронированиями
* Просмотр доступных автомобилей
* Отслеживание статуса заказов
* Учет автомобилей
* Контроль за состоянием машин
* Управление персоналом
* Анализ данных
* Интеграция с другими системами

**2.4. СООБЩЕНИЯ ОПЕРАТОРУ**

В процессе выполнения программы могут возникать различные сообщения, информирующие оператора о состоянии работы системы.

Сообщения об ошибках могут указывать на проблемы с доступом к данным, неправильные входные данные или другие причины, мешающие выполнению программы.

Информационные сообщения могут содержать информацию о новых функциях программы, обновлениях и других важных событиях.

В случае ввода неправильного логина или пароля, информационная система «Прокат автомобиле й» выдаст сообщение об ошибке. Оператор увидит сообщение, указывающее на то, что логин или пароль введены неверно.



Рис. 51 Сообщение об ошибке

Если оператор нажмет на кнопку «Печать» в информационной системе «Прокат автомобилей», а принтер при этом не будет подключен, система выдаст сообщение об ошибке. Сообщение может содержать информацию о том, что принтер не подключен или не доступен.



Рис. 52 Сообщение о принтере

В случае если оператор нажмет на кнопку «Выход из аккаунта», система выдаст сообщение для подтверждения своих действий.



Рис. 53 Сообщение подтверждения выхода из аккаунта

Подобная функция также работает в случае если оператор нажмет на кнопку «Выход», находясь при этом в окне авторизации. Система выдаст сообщение для подтверждения своих действий.



Рис. 54 Сообщение подтверждения выхода из окна авторизации

Также в случае если оператор нажмет на кнопку «Выход из приложения», система выдаст ему сообщение для подтверждения своих действий.



Рис. 55 Сообщение подтверждения выхода из приложения