**САНКТ-ПЕТЕРБУРГСКОЕ ГОСУДАРСТВЕННОЕ**

**БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»**

**ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ**

МДК 01.01. Разработка программных модулей

Специальность 09.02.07

«Информационные системы и программирование»

Квалификация «Программист»

Руководитель учебной практики:

Матысик И.А.

Выполнил студент группы 493:

Кашицын А.А.

Санкт-Петербург, 2023

ОГЛАВЛЕНИЕ

[ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ 3](#_Toc128101648)

[ВВЕДЕНИЕ 5](#_Toc128101649)

[1. ОБЩАЯ ЧАСТЬ 8](#_Toc128101650)

[1.1 Цель разработки 8](#_Toc128101651)

[1.2 Описание предметной области 8](#_Toc128101652)

[1.3 Анализ средств и технологий 9](#_Toc128101653)

[2. ПРОЕКТНАЯ ЧАСТЬ 12](#_Toc128101654)

[2.1. Постановка задачи 12](#_Toc128101655)

[2.2. Функционально-логическая структура проекта 13](#_Toc128101656)

[2.2.1. Структурная схема приложения 14](#_Toc128101657)

[2.2.2. Диаграмма вариантов использования 15](#_Toc128101658)

[2.2.3. Диаграмма деятельности 16](#_Toc128101659)

[2.3. Логическая модель данных 17](#_Toc128101660)

[2.4. Разработка программных модулей 19](#_Toc128101661)

[2.4.1. Пользовательский интерфейс 20](#_Toc128101662)

[2.5. Руководство пользователя 20](#_Toc128101663)

[2.6. Выводы по учебной практике 25](#_Toc128101664)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 27](#_Toc128101665)

[ПРИЛОЖЕНИЕ 1. Исходный программный код модуля «Авторизация» 29](#_Toc128101666)

[ПРИЛОЖЕНИЕ 2. Исходный программный код модуля «Создание пользователя» 32](#_Toc128101667)

[ПРИЛОЖЕНИЕ 3. Структура репозитория 35](#_Toc128101668)

[ПРИЛОЖЕНИЕ 4. Структура проекта 36](#_Toc128101669)

# **ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ**

Требуется разработать информационную систему (ИС) «MyYouGile» – систему управления проектами. Элементом в данной системе является проект. Для этого требуется разработать основные программные модули ИС для следующих пользователей:

* + разработчик проекта может записываться на проект;
  + заказчик проекта может создавать, редактировать, удалять проекты, менять статус проекта;
  + менеджер проектов может назначать разработчика проекту

Выполнить анализ и проектирование бизнес-процессов разрабатываемой информационной системы путем составления диаграммы прецедентов (Use Case) для пользователей системы и диаграммы деятельности для прецедента «Изменение статуса проекта».

Выполнить проектирование реляционного хранилища данных путем составления ER-диаграммы разрабатываемой информационной системы и ведение словаря данных.

Разработать базу данных информационной системы на сервере баз данных, создать таблицы основных сущностей, атрибуты, отношения и необходимые ограничения. Выполнить импорт подготовленных данных в разработанную базу данных информационной системы.

Выполнить разработку desktop-приложения. Создать модуль авторизации – первое окно, что видит пользователь, на которой пользователю предлагается ввести свой логин и пароль. Только после успешной авторизации пользователь получает доступ к остальным модулям системы.

После авторизации пользователь под любой ролью сначала попадает на экран просмотра проектов, которые хранятся в базе данных. При отсутствии изображения необходимо вывести картинку-заглушку.

Пользователь должен иметь возможность искать проекты по названию, сортировать проекты по возрастанию и убыванию их стоимости.

Кроме этого, пользователь должен иметь возможность отфильтровать данные по заказчику проекта. Первым элементом в списке заказчиков должен быть «Все заказчики», при выборе которого настройки фильтра сбрасываются.

В верхней части окна необходимо показывать количество выведенных данных и общее количество записей в базе.

Необходимо реализовать возможность добавления и редактирования проектов. Редактировать данные может только заказчик.

Редактирование и добавление проекта выполняется в новом окне. Переходы на данное окно должны быть реализованы из формы списка проектов: для редактирования – при нажатии кнопки «Добавить проект».

ID проекта при добавлении не отображается, автоматически вычисляется инкрементом к имеющемуся в базе данных, при редактировании ID доступно только для чтения.

Пользователь может добавить и заменить изображение логотипа проекта.

В окне редактирования проекта должна присутствовать кнопка «Удалить», которая удаляет проект из базы данных. Проект, на который записан разработчик, удалить нельзя.

После редактирования, добавления, удаления проектов, данные в окне списка проектов должны быть обновлены.

Должна быть реализована возможность формирования заказа на участие в разработке проекта из окна просмотра списка проектов для авторизованных пользователей системы «MyYouGile».

При просмотре списка проектов пользователь может выделять проект, по нажатию правой клавиши мыши вызвать контекстное меню с командой «Выбрать». При нажатии на команду к данному проекту приписывается пользователь. При этом в интерфейсе появляется возможность задать условия реализации проекта: сроки и основные этапы.

Разрабатываемая система направлена на организацию поиска проектов для исполнителей, исполнителей для соискателей.

# **ВВЕДЕНИЕ**

Учебная практика по МДК 01.01 «Разработка программных модулей» является обязательным учебным элементом для формирования общих и профессиональных компетенцией по модулю ПМ 01 «Разработка модулей программного обеспечения для компьютерных систем».

Практика проходит в колледже информационных технологий, в компьютерной лаборатории №213 «Прикладное программирование». Период прохождения практики с 23.02.2022 по 15.03.2022. Объем учебной практики 30 часов. Тема учебной практики: «Разработка информационной системы отделения банка по кредитованию клиентов».

В отчете к учебной практике представлена краткая необходимая теоретическая справка, представлена четкая постановка задачи, представлены диаграммы, описывающие функциональную структуру программы и логическую модель данных, подробная инструкция пользователя и выводы по проделанной работе. Представлен программный код в приложении. В результате прохождения учебной практики по МДК 01.01 были подтверждены общие и профессиональные компетенции.

Перечень общих компетенций:

ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 4. Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 5. Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 6. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения.

ОК 7. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.

ОК 8. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.

ОК 9. Использовать информационные технологии в профессиональной деятельности.

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках.

Перечень профессиональных компетенций:

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

В результате прохождения учебной практики по МДК 01.01 «Разработка программных модулей», был получен практический опыт в:

* + в разработке кода программного продукта на основе готовой спецификации на уровне модуля;
  + в использовании инструментальных средств на этапе отладки программного продукта.

В результате прохождения учебной практики, были получены умения:

* + осуществлять разработку кода программного модуля на языке высокого уровня;
  + создавать программу по разработанному алгоритму как отдельный модуль;
  + осуществлять разработку кода программного модуля на современных языках программирования;
  + выполнять оптимизацию и рефакторинг программного кода;
  + оформлять документацию на программные средства;
  + разрабатывать код программного модуля с применением технологии объектно-ориентированного программирования;
  + использовать современные технологии обработки данных.

В результате прохождения учебной практики по модулю, были получены знания по следующим понятиям:

* + основные этапы разработки программного обеспечения;
  + основные принципы технологии структурного и объектно-ориентированного программирования;
  + основные возможности современных технологий.

# **ОБЩАЯ ЧАСТЬ**

В данной части отчета по учебной практике представлена краткая теория по теме учебной практике, а также представлены результаты исследования проведенного анализа программных средств и технологий для разработки проекта учебной практики.

## **Цель разработки**

Целью проекта является проектирование и разработка информационной системы для специалиста кредитного отдела банка, которая поможет автоматизировать процесс кредитования, что сократит время обслуживания клиента банка. В итоге разрабатываемая система должна привлечь больше потенциальных клиентов банка, обеспечив стабильный рост доходов и рейтингов банка, для которого разрабатывается данная информационная система.

## **Описание предметной области**

Информационная система "Кредитный отдел" – прикладное программное обеспечение, использующееся для кредитования физических лиц, просмотра информации по кредиту, а также предназначена для решения следующих задач:

* + контроль документооборота;
  + формирование необходимой информации по итогам заведения кредита на физическое или юридическое лицо (дата выдачи кредита, количество месяцев кредитования, необходимая сумма для погашения долга за месяц).

Основными целями внедрения системы являются:

* + упрощение контроля кредитования;
  + полный контроль над статусом кредита;
  + минимизация человеческого фактора при ведении документов;
  + снижение рутиной работы при работе с документами.

Исходная информация в систему поступает из документов о заёмщике, кредитного договора. Входными данными являются следующая информация:

* + дата и номер кредитного договора;
  + сведения о кредиторе;
  + сведения о заёмщике;
  + срок кредита.

Рассмотрим определение прецедентов (вариантов использования). Система требуется, прежде всего, следующим заинтересованным лицам:

* + работник банка;
  + клиент;
  + системный администратор.

Функционал каждой роли:

Системный администратор и работник банка обладают возможностью просмотра, редактирования, добавления, удаления данных. Ко всему прочему системный администратор имеет возможность создавать пользователей для системы, удалять их или же изменять их учётные данные.

Клиент может только просматривать данные.

Система должна функционировать в многопользовательском режиме, поэтому каждый пользователь должен иметь свой пароль доступа в систему.

Проект предназначен для коммерческого банка, занимающегося выдачей кредитов физическим и юридическим лицам.

Внедрение данного проекта поможет автоматизировать процесс добавления новых клиентов, способствуя снижению рутиной работы с ведением документооборота и минимизации случаев человеческого фактора.

## **Анализ средств и технологий**

В данном разделе описываются возможные пути и программные средства, с помощью которых можно реализовать приложение.

В данном подразделе пояснительной записки представлен материал проведенного анализа программных средств, которыми можно разработать приложение. В анализе присутствуют сравнительные характеристики программных технологий и методов, языков программирования и интегрированных сред.

Инструментальные средства:

* + операционная система Windows 10;
  + язык программирования C#;
  + среда программирования Visual Studio 2019;
  + СУБД SQL Server Management Studio.

Для создания приложения будет использована интегрированная среда программирования Microsoft Visual Studio, язык программирования высокого уровня С#, систему построения клиентских приложений Windows Forms, а также SQL Server Management для создания базы данных.

**Microsoft Visual Studio**

Инструментальная среда разработки, включающая в себя интегрированную среду разработки, редактор исходного кода, встроенный отладчик. Многие другие инструменты возможно получить благодаря подключению плагинов – сторонних расширений. Был выбран по причине поддержки языка программирования С#. Также это наиболее изученная разработчиком платформа, что позволяет быстрее разрабатывать приложения не тратя время на изучения нового материала.

**C#**

Объектно-ориентированный язык программирования. Разработан в 1998 — 2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework.

Достоинства С# по книге Биллига:

* + создавался параллельно с каркасом Framework .Net и в полной мере учитывает все его возможности - как FCL, так и CLR;
  + является полностью объектно-ориентированным языком, где даже типы, встроенные в язык, представлены классами;
  + является мощным объектным языком с возможностями наследования и универсализации;
  + является наследником языков C/C++, сохраняя лучшие черты этих популярных языков программирования. Общий с этими языками синтаксис, знакомые операторы языка облегчают переход от С++ к C#;
  + сохранив основные черты своего великого родителя, язык стал проще и надежнее. Простота и надежность, главным образом, связаны с тем, что на C# хотя и допускаются, но не поощряются такие опасные свойства С++ как указатели, адресация, разыменование, адресная арифметика;
  + благодаря каркасу Framework .Net, ставшему надстройкой над операционной системой, программисты C# получают те же преимущества работы с виртуальной машиной, что и программисты Java;
  + мощная библиотека каркаса поддерживает удобство построения различных типов приложений на C#, позволяя легко строить Web-службы, другие виды компонентов, достаточно просто сохранять и получать информацию из базы данных и других хранилищ данных;
  + реализация, сочетающая построение надежного и эффективного кода, является немаловажным фактором, способствующим успеху C#.

Выделение и объединение лучших идей современных языков программирования делает язык C# не просто суммой их достоинств, а языком программирования нового поколения.

**Windows Forms**

Windows Forms — интерфейс программирования приложений, отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework. Данный интерфейс упрощает доступ к элементам интерфейса Microsoft Windows за счет создания обёртки для существующего Win32 API в управляемом коде.

**SQL Server Management Studio**

Утилита из Microsoft SQL Server 2005 и более поздних версий для конфигурирования, управления и администрирования всех компонентов Microsoft SQL Server. Утилита включает скриптовый редактор и графическую программу, которая работает с объектами и настройками сервера.

# **ПРОЕКТНАЯ ЧАСТЬ**

В данной части отчета по учебной практике представлено подробное описание разработанного приложения, соответствующего поставленной задаче в техническом задании к учебной практике.

## **Постановка задачи**

Информационная система "Отдел кредитования" – прикладное программное обеспечение, использующееся для кредитования физических лиц, просмотра информации по кредиту, а также предназначена для решения следующих задач:

* + контроль документооборота;
  + формирование необходимой информации по итогам заведения кредита на физическое или юридическое лицо (дата выдачи кредита, количество месяцев кредитования, необходимая сумма для погашения долга за месяц).

Основными целями внедрения системы являются:

* + упрощение контроля кредитования;
  + полный контроль над статусом кредита;
  + минимизация человеческого фактора при ведении документов;
  + снижение рутиной работы при работе с документами.

Исходная информация в систему поступает из документов о заёмщике, кредитного договора. Входными данными являются следующие:

* + дата и номер кредитного договора;
  + сведения о кредиторе;
  + сведения о заёмщике;
  + срок кредита.

Рассмотрим определение прецедентов (вариантов использования). Система требуется прежде всего следующим заинтересованным лицам:

* + работник банка;
  + клиент;
  + системный администратор.

Определяется перечень функциональных подсистем, их назначение и основные характеристики, требования к числу уровней иерархии и степени централизации системы.

Смежными системами для «Клиенты» являются:

* + информационные системы ввода, хранения, поиска и оперативной обработки информации о клиентах;
  + информационная система формирование отчетов, необходимых работнику банка, содержащих все данные о клиенте: его адрес, ФИО, номер телефона, ИНН, паспортные данные и номер кредитного договора.

Источниками данных для Системы должны быть:

* + СУБД MS SQL;
  + таблицы Excel.

Входными данными являются следующие:

* + дата заключения договора;
  + номер документа;
  + паспорт и ИНН;
  + срок кредитования;
  + сумма кредита.

Выходными данными являются следующие виды отчетов:

* + распечатанный договор о заключении кредита.

Система должна функционировать в многопользовательском режиме и давать возможность:

* + просматривать записи базы данных, в том числе и с помощью различных фильтров;
  + добавлять новые записи и изменять существующие;
  + удалять записи.

## **Функционально-логическая структура проекта**

В данном разделе отчета по учебной практике представлены различные диаграммы на языке UML, описывающие функциональную и логическую структуру разрабатываемой программы учебной практики.

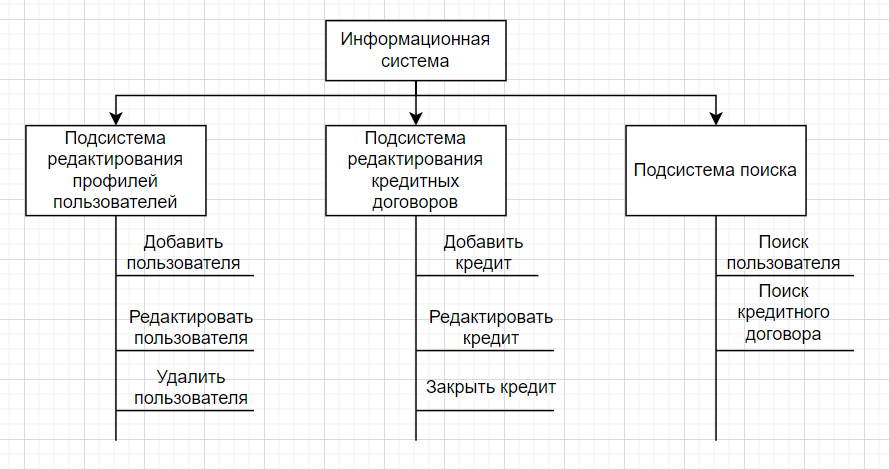


Рисунок 1 – Структурная схема проекта

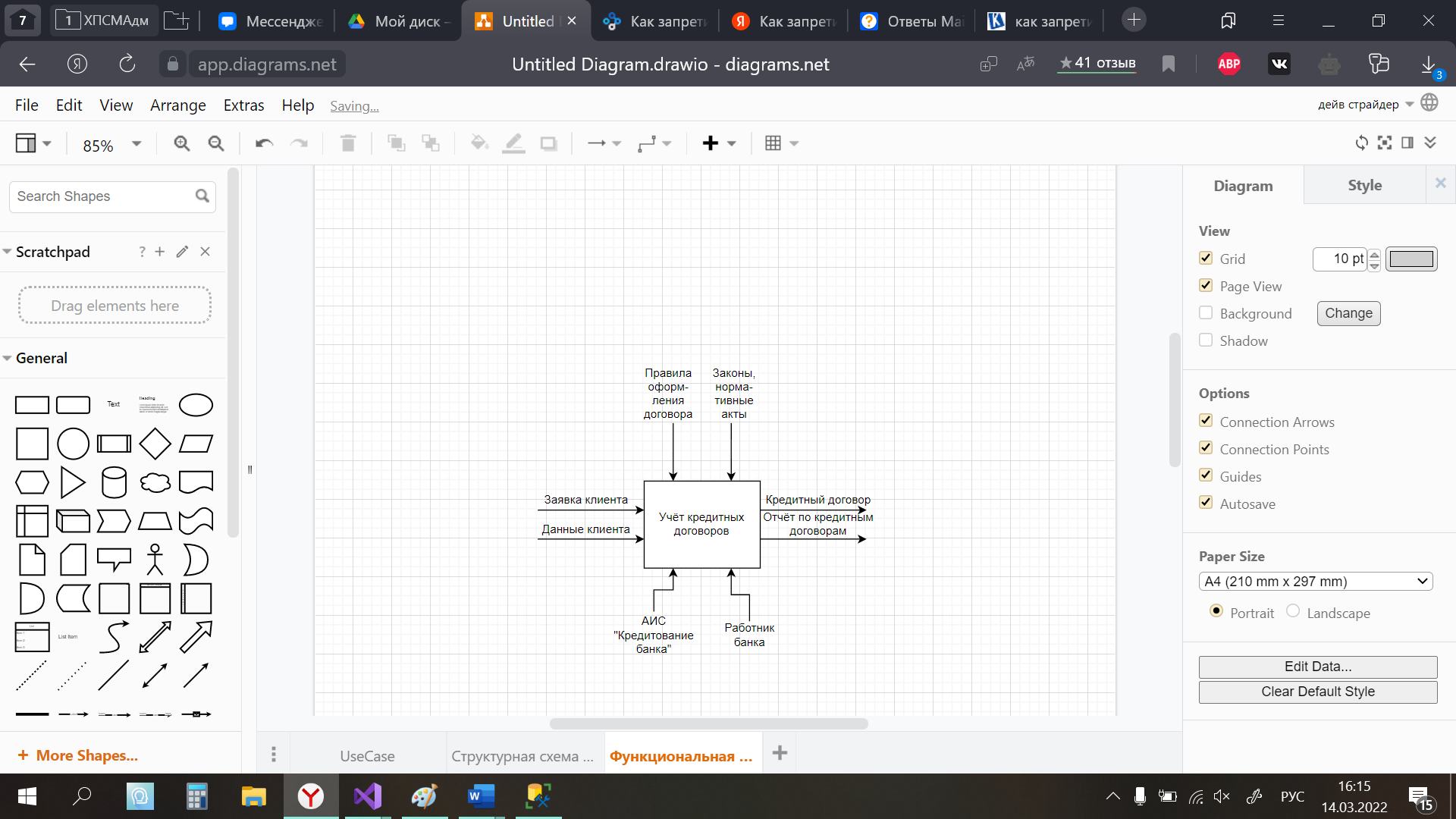


Рисунок 2 – Функциональная диаграмма начального уровня

### **Структурная схема приложения**

Структурная схема приложения используется для того, чтобы графически обозначить взаимосвязи между объектами системы.

В следующей структурной схеме приведены формы, которые используют в приложении, а также их утилитарная функция, красным выделены переходы между формами, как показано на рисунке 3.

Форма авторизации предназначена для авторизации пользователей в системе.

Форма системного администратора предназначена для создания, удаления и просмотра пользователей в системе, а также их редактирования.

Форма работника банка предназначена для добавления новых клиентов, просмотра информации клиента и просмотра статуса его кредита.

Форма личного кабинета предназначена для просмотра клиентом своего профиля и отслеживания статуса кредита.

Форма работы с клиентом предназначена для добавления клиента и редактирования его профиля.

Форма для работы с кредитом предназначена для просмотра кредитного договора и его редактирования.

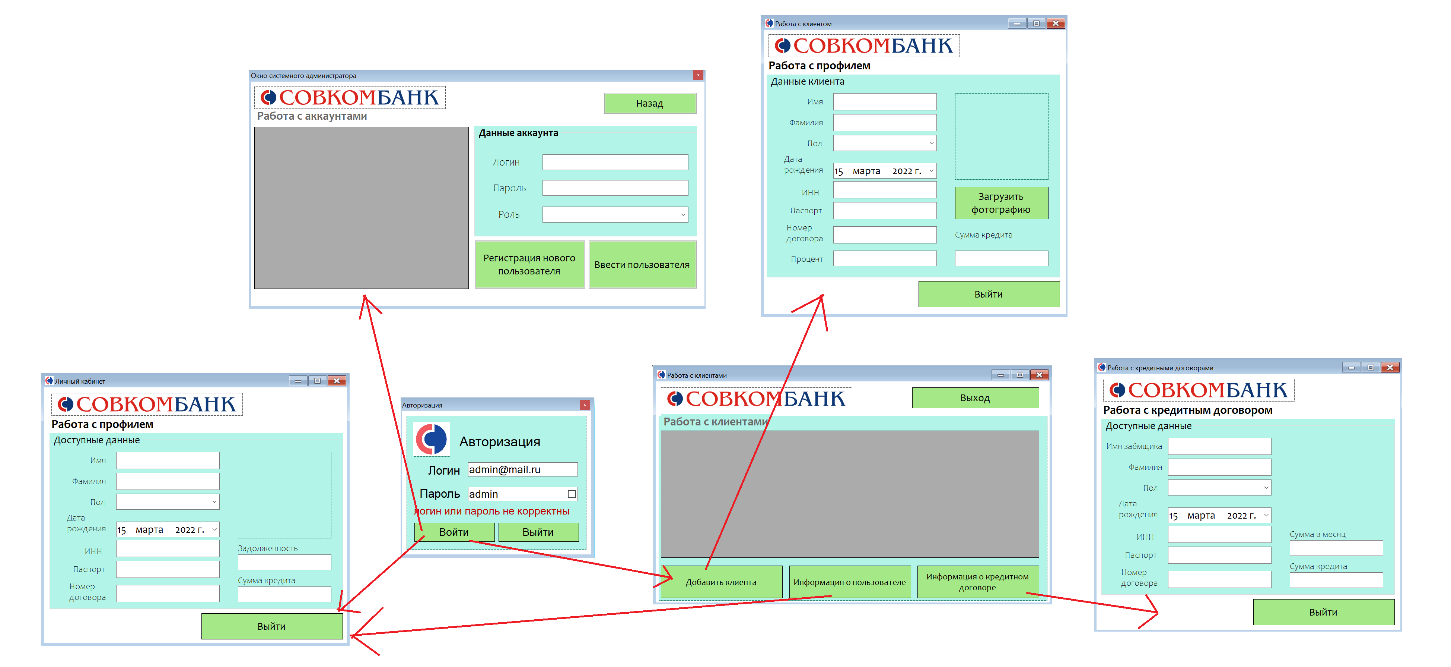


Рисунок 3 – Структурная схема приложения

### **Диаграмма вариантов использования**

В этом разделе приведена UML-диаграмма типа вариантов использования или диаграмма прецедентов, представленная на рисунке 4. Диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне. Прецедент - возможность моделируемой системы, благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат.

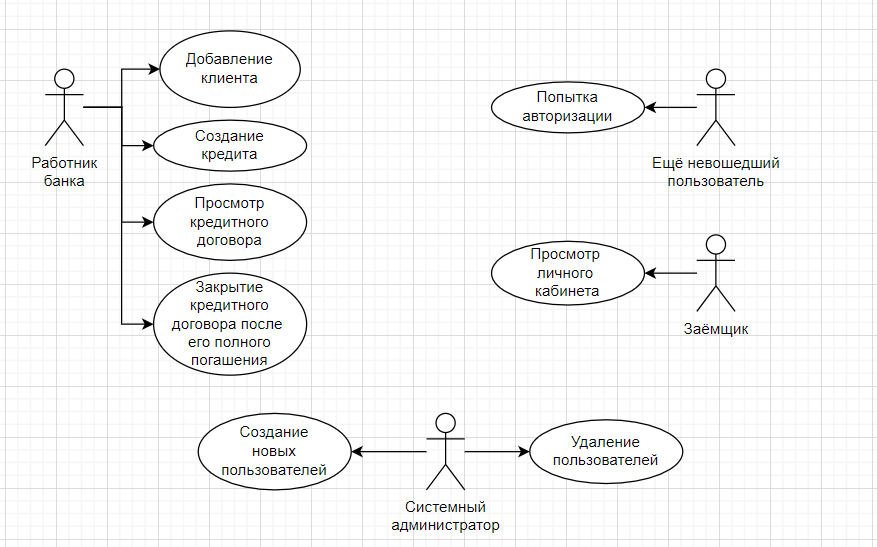


Рисунок 4 – Диаграмма вариантов использования

### **Диаграмма деятельности**

Диаграмма деятельности (англ. activity diagram) – UML-диаграмма, на которой показаны действия, состояния которых описаны на диаграмме состояний. Под деятельностью понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов – вложенных видов деятельности и отдельных действий, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого.

Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.

Была разработана диаграмма деятельности для прецедента «Добавление нового пользователя», которая осуществляется ролью системный администратор. Данная диаграмма представлена на рисунке 5.

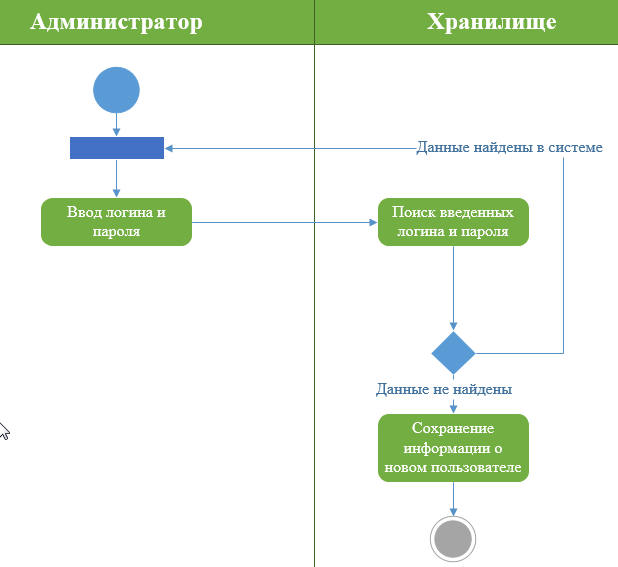


Рисунок 5 – Диаграмма деятельности

## **Логическая модель данных**

В основе разрабатываемого приложения находится база данных, как структурированное хранилище информации разрабатываемой системы. Удобно логическую модель данных описывать с применением UML-языка на основе ED-диаграммой. На рисунке 6 представлена ER-диаграмма разработанной базы данных.

В системе есть две сущности: «Заёмщики» и «Кредиты».

«Заёмщики» состоят из таких столбцов, как их ID, Имя, Фамилия, Отчество, Пол, Дата Рождения, данные паспорта и ИНН, а также ID выданного на их имя кредита в банке.

«Кредиты» состоят из таких столбцов, как ID, IDUser, Номер кредитного договора, Срок Кредитования, Сумма Кредита, Процент и Оплаченная часть.

Таблица «BDGenders» является таблицей-справкой и состоит из таких столбцов, как ID и Title, в котором хранятся данные о поле клиента: мужской или женский.

Таблица «BDRole» является таблицей-справкой и состоит из таких столбцов, как ID и Title, в котором хранятся данные о роли пользователя: администратор, работник банка и клиент.

Таблица «BDUsers» состоит из таких столбцов, как ID, IDRole, Email и Password и описывает данные для входа в систему.

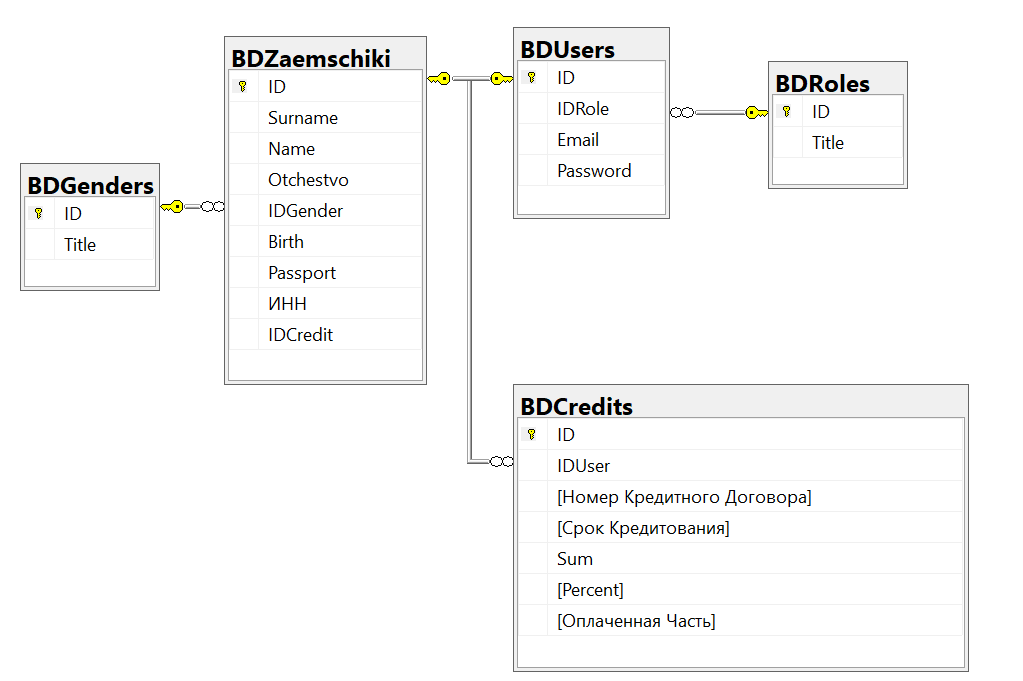


Рисунок 6 – ER-диаграмма

На рисунке 7 представлен заполненный словарь данных системы – набор информации, описывающий, какой тип данных хранится в БД, их формат, структуру и способы использования данных. Дополнительно в словаре данных могут содержаться описания к отдельным полям таблицы базы данных.

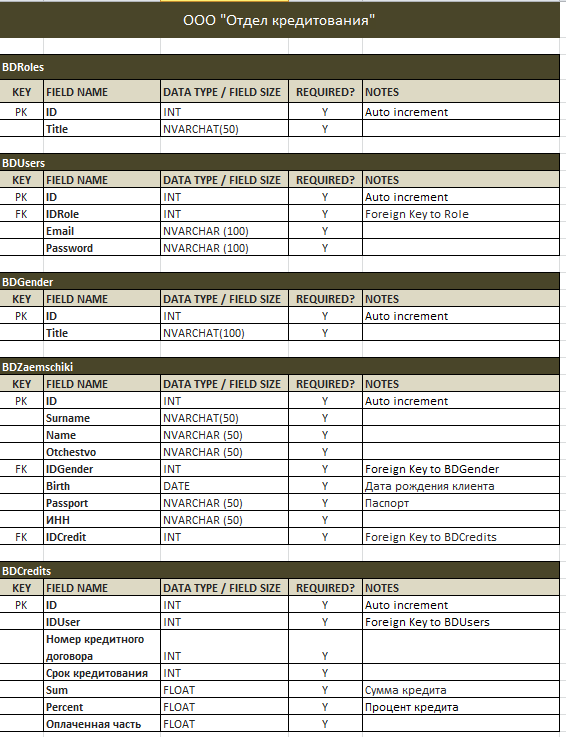


Рисунок 7 – Словарь данных системы

## **Разработка программных модулей**

В данном разделе отчета представлены действия, необходимые для программной реализации разрабатываемого приложения, а так же требования к разработке пользовательского интерфейса.

### **Пользовательский интерфейс**

Типы требований к пользовательскому интерфейсу:

* + в качестве основного фона используется белый цвет; в качестве дополнительного: #B3F4E9;
  + для акцентирования внимания пользователя на целевое действие интерфейса используется цвет #A5E887.

|  |  |  |
| --- | --- | --- |
| Основной цвет | Дополнительный цвет | Акцентирование  внимания |
| #FFFFFF | #B3F4E9 | #A5E887 |
|  |  |  |

* + используется шрифт Condar.

Программа не раскрывается на весь экран и не масштабируется.

Кнопки приложения имеют цвет #A5E887, так же при наведении курсор мыши принимает форму руки. Все элементы форм сгруппированы в панель.

Не допускаются орфографические и грамматические ошибки.

Все экранные формы пользовательского интерфейса должны иметь заголовок с логотипом logo.png (расположен в ресурсах). Нельзя искажать логотип (не изменяйте изображение, его пропорции, цвет).

Также для приложений должна быть установлена иконка.

В случае возникновения ошибок система оповещает пользователя всплывающими окнами с информацией об ошибке.

## **Руководство пользователя**

Для начала работы запускаем ярлык – BancCredits.exe.

**Форма ввода данных для входа**

После запуска программы открывается окно аутентификации, где нужно ввести логин и пароль для входа в систему, на рисунке 8 продемонстрировано данное окно.

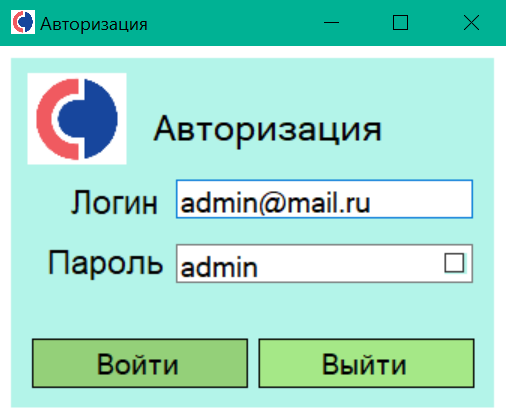


Рисунок 8 – Форма для ввода данных

При ошибочном вводе пароля или логина вам будет показано сообщение о том, что вы должны проверить правильность ввода, как показано на рисунке 9. Сообщение было показано, отслеживание ошибок ввода пароля работает.

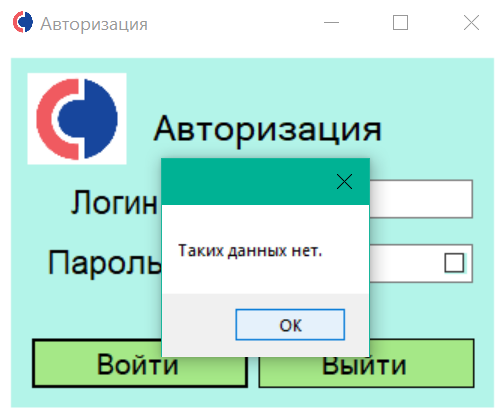


Рисунок 9 – Результат неправильного ввода пароля

В случае корректного ввода данных, программа сразу же откроет окно пользователя, под которым вы зашли. В данном случае это администратор, как показано на рисунке 10. При правильном пароле была открыта форма администратора, тестирование аутентификация прошло успешно.

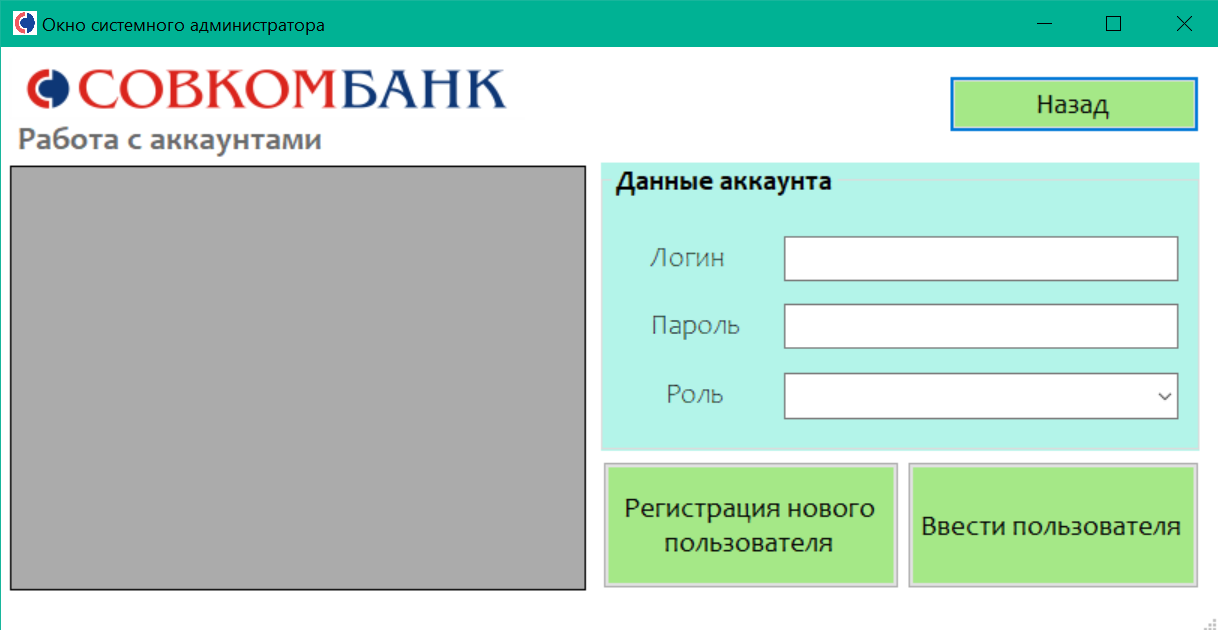


Рисунок 10 – Успешная авторизация

**Форма системного администратора**

Форма находится в центре экрана, на рисунке 11 продемонстрирован её внешний вид. При введении данных в TextBox и нажатии кнопки «Регистрация нового пользователя» – пользователь заносится в таблицу. При нажатии «Ввести пользователя» он подсвечивается в таблице пользователей.

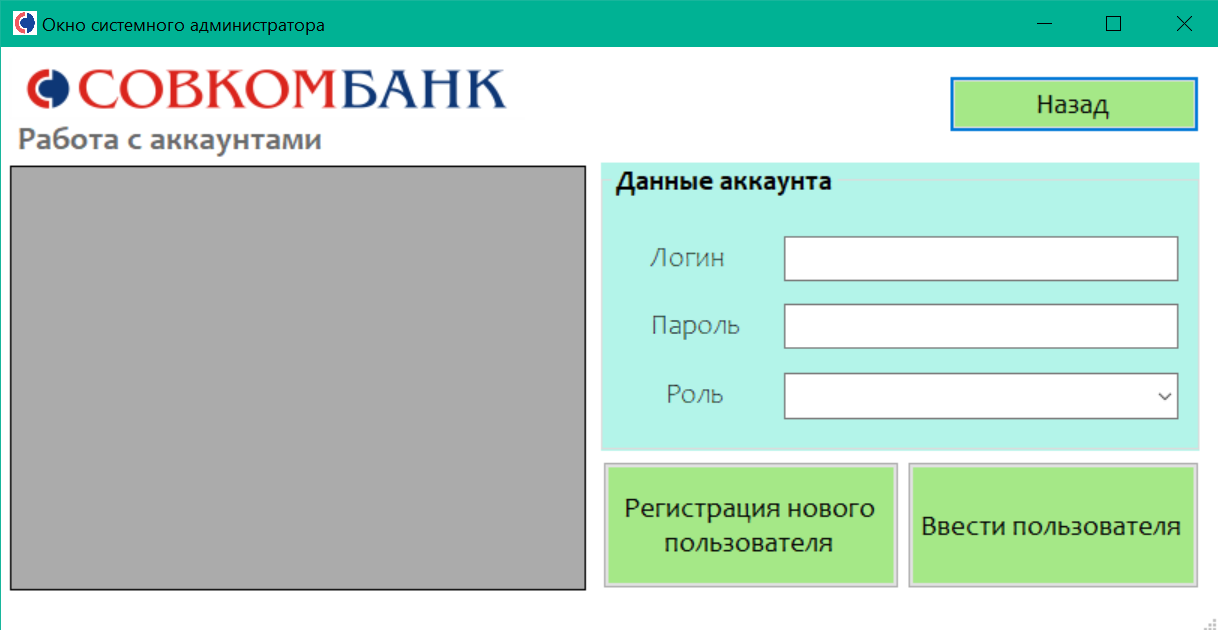


Рисунок 11 – Форма системного администратора

**Форма работы с клиентами**

При входе в качестве «Работника банка» открывается форма работы с клиентами, как показано на рисунке 12.

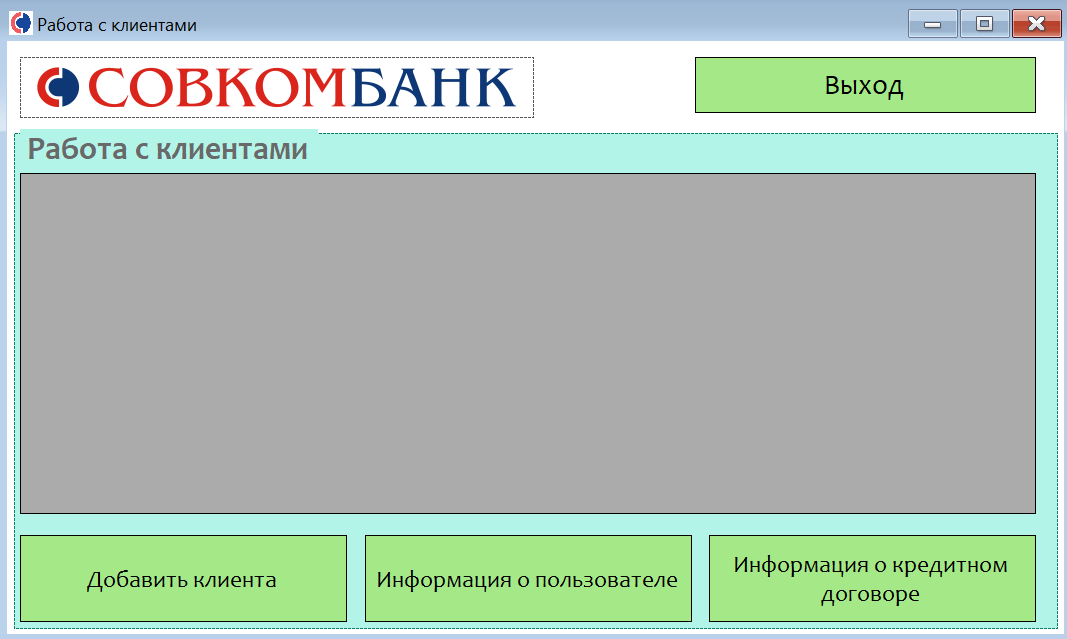


Рисунок 12 – Форма работы с клиентами

**Форма создания/регистрации нового пользователя**

При нажатии на кнопку «Добавить клиента» открывается форма работы с клиентом, в которой заполняются данные о клиенте, и загружается его фотография. Пример работы данной формы показа на рисунке 13.

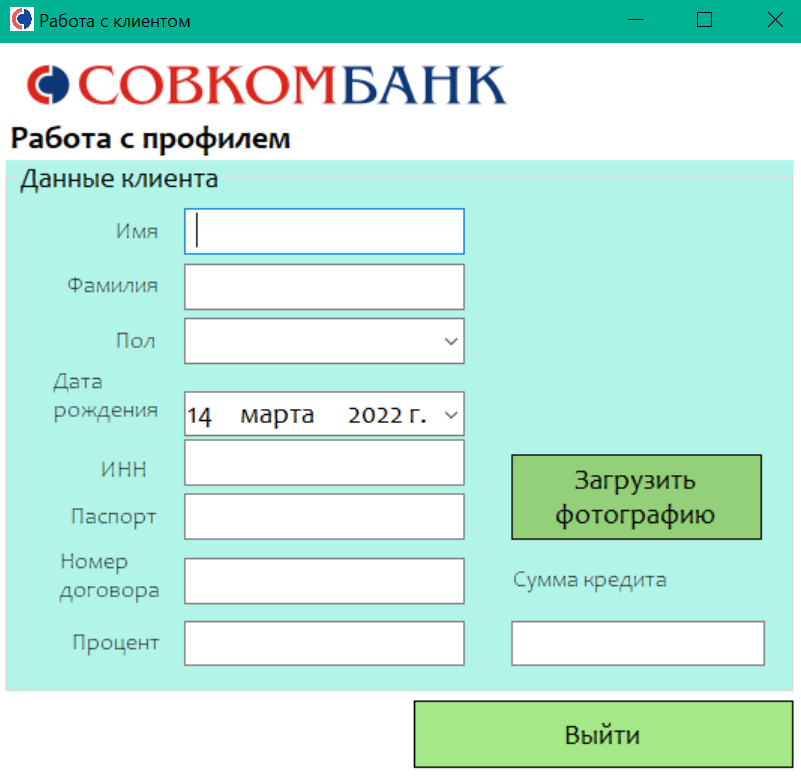


Рисунок 13 – Форма работы с клиентом

**Форма просмотра информации клиента**

При нажатии на кнопку «Ввести пользователя» открывается форма работы с клиентом, в которой просматривается информация пользователя. Эта же форма открывается, когда клиент входит под своими данными. Пример данной формы представлен на рисунке 14.

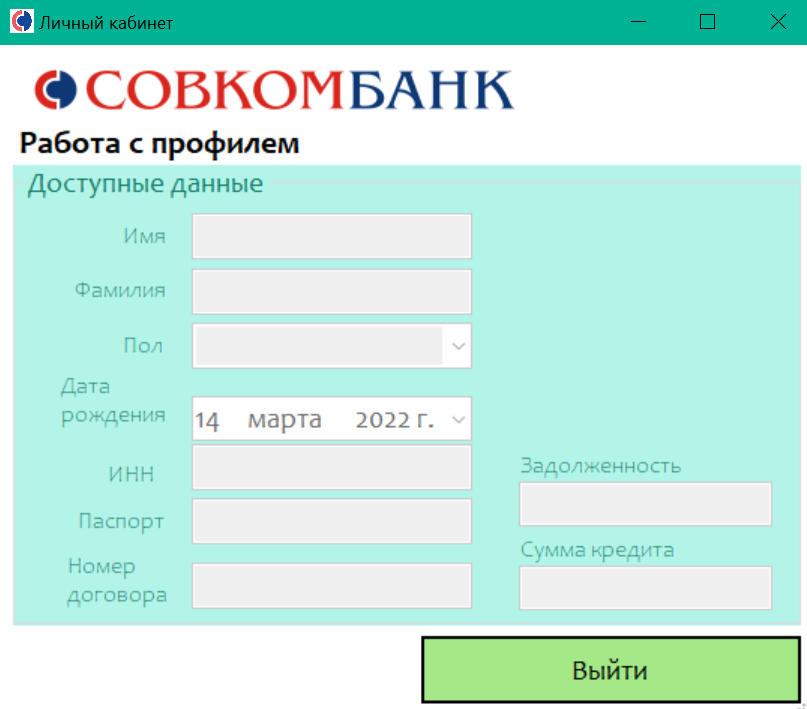


Рисунок 14 – Форма просмотра информации клиента

**Форма работы с кредитными договорами**

При нажатии кнопки «Информация о кредитном договоре» в форме работы с клиентами открывается форма работы с кредитными договорами. Пример данной формы представлен на рисунке 15.

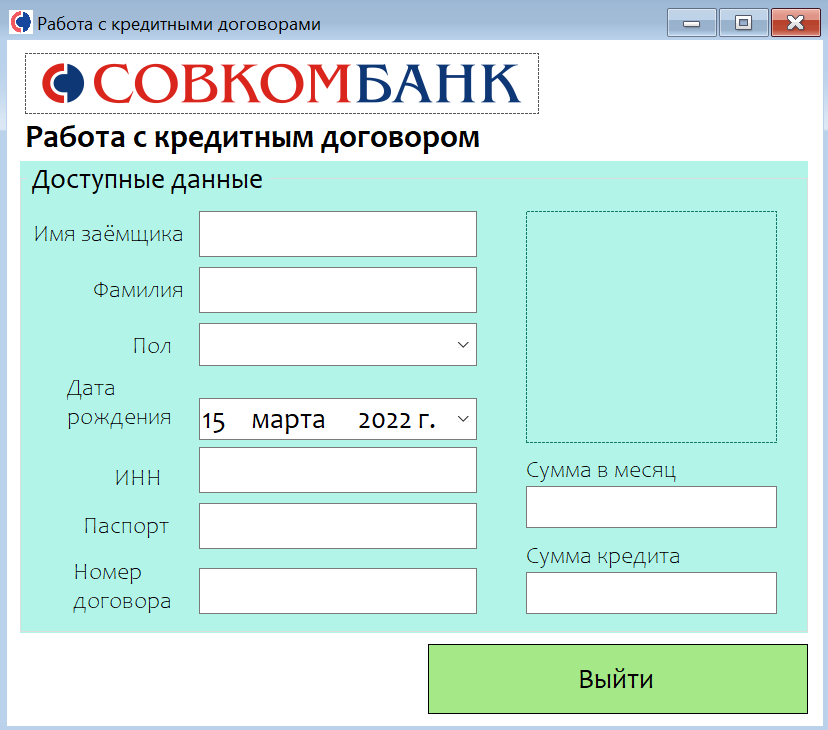


Рисунок 15 – Форма работы с кредитными договорами

## **Выводы по учебной практике**

Темой учебной практики связана с разработкой автоматизированной информационной системы отделения банка по кредитованию клиентов.

Было выполнено:

* + анализ предметной области;
  + выбор технологий и средств;
  + создана функционально-логическая структура проекта;
  + разработан пользовательский интерфейс;
  + были построены диаграммы проекта;
  + реализованы некоторые программные модули, такие как «Авторизация».

Учебная практика проходила в конце обучения на четвертом курсе и подтвердила приобретенные в ходе учебного процесса профессиональные и общие компетенции по МДК 01.01 «Разработка программных модулей».

В ходе выполнения учебной практике разрабатывалось прикладное приложение для автоматизации кредитования банка.

Достоинством данного приложения является его простота использования пользователем, т.к. оно имеет интуитивно понятный стандартизированный интерфейс.

Благодаря данной учебной практике были закреплены полученные знания и практические умения при изучении МДК 01.01 «Разработка программных модулей», более глобально изучена среда программирования Visual Studio, язык программирования C#, технология разработки и программирования баз данных.

В итоге был, получен опыт в работе по данной теме, так же были освоены основные приемы работы с базами данных в среде программирования Visual Studio 19, применена технология ADO.Net, работающая с базами данных. Важным аспектом, было изучение новых компонентов, с помощью которых осуществлялась связь и отображение информации базы данных.

Были реализованы основные способы манипуляции с базами данных, а именно: просмотр, редактирование, поиск и фильтрация по заданным критериям.

База данных, кроме числовых и строковых полей, позволяет хранить и использовать двоичные поля, которые представляются изображениями отдельных записей. Программа снабжена простым интерфейсом, но, тем не менее, были реализованы все необходимые операции с таблицами баз данных.

# **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Г. Н. Федорова Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: учебное пособие / Г. Н. Федорова. – Москва : КУРС : ИНФРА-М, 2020 – 336 с. (Среднее профессиональное образование).
2. Гагарина Л. Г. Технология разработки программного обеспечения : учебное пособие / Л. Г. Гагарина, Е. В. Кокорина, Б. Д. Сидорова-Виснадул : под ред. Л. Г. Гагариной. – Москва : ИД «ФОРУМ» : ИНФРА-М, 2021. – 400 с. – (Среднее профессиональное образование).
3. Федорова Г. Н. Осуществление интеграции программных модулей: учебник для студ. учреждений сред. проф. образования / Г. Н. Федорова. 2-е изд., стер. – М. : Издательский центр «Академия», 2018. – 288 с.
4. Перлова О. Н. Проектирование и разработка информационных систем : учебник для студ. учреждений сред. проф. образования / О. П. Ляпина, А. В. Гусева. - 3-е изд., испр. – М. : Издательский центр «Академия», 2020. – 256 с.
5. Зверева В.П. Сопровождение и обслуживание программного обеспечения компьютерных систем : учебник для студ. учреждений сред. проф. образования / В. П. Зверева, А. В. Назаров. – 2-е изд., испр. – М. : Издательский центр «Академия», 2020. -256 с.
6. Голицына О.Л. Основы алгоритмизации и программирования : учебное пособие / О.Л. Голицына, И.И. Попов. — 4-е изд., испр. и доп. — Москва : ФОРУМ : ИНФРА-М, 2021. — 431 с. — (Среднее профессиональное образование).
7. Хорев П.Б. Объектно-ориентированное программирование с примерами на С# : учебное пособие / П.Б. Хорев. — Москва : ФОРУМ : ИНФРА-М, 2021. — 200 с. — (Среднее профессиональное образование).
8. Гуриков С. Р. Введение в программирование на языке Visual C# : учеб. пособие / С.Р. Гуриков. — М. : ФОРУМ : ИНФРА-М, 2019. — 447 с. — (Среднее профессиональное образование).
9. 3 Зубкова Т. М. Федорова технология разработки программного обеспечения: учебное пособие / Т. М. Зубкова. – Санкт-Петербург: Лань, 2019. - 324 с.: ил.
10. <https://metanit.com/> – метанит
11. <https://docs.microsoft.com/ru-ru/> – техническая документация от компании Майкрософт
12. <https://habr.com> – харб
13. <https://nationalteam.worldskills.ru/skills/programmnye-resheniya-dlya-biznesa/> – сайт WorldSkills.

# **ПРИЛОЖЕНИЕ 1. Исходный программный код модуля «Авторизация»**

using System;

using System.Threading;

using System.Windows.Forms;

namespace WorldSmartphones

{

public partial class FormAuth : Form

{

public FormAuth()

{

InitializeComponent();

}

static int x = 1;

static int y = 0;

public int idrole;

ClassUsers users = new ClassUsers();

World\_smartphonesDataSet1.UsersDataTable dataUsers;

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

{

dataUsers = this.usersTableAdapter1.GetData();

int totalCount = dataUsers.Count;

}

public void buttonLogin\_Click(object sender, EventArgs e)

{

users.email = textBoxEmail.Text;

users.password = textBoxPass.Text;

users.AuthorizationUsers();

}

public void AuthorizationRolesUsers()

{

string log, pas, stringtS;

pas = textBoxPass.Text;

log = textBoxEmail.Text;

dataUsers = this.usersTableAdapter1.GetData();

try

{

if (x < 3)

{

switch (idrole)

{

case 0:

MessageBox.Show("Вы неверно ввели логин (и)или пароль");

x++;

break;

case 1:

MessageBox.Show("Приветствуем, огранизатор");

FormOrg fO = new FormOrg();

Hide();

fO.ShowDialog();

break;

case 2:

MessageBox.Show("Приветствуем, продавец");

FormProdawec fPr = new FormProdawec();

this.Hide();

fPr.ShowDialog();

break;

}

}

else if (x == 3)

{

if (y == 0)

{

x = 0;

y++;

MessageBox.Show("Был введен неверный логин/пароль 3 раза,

подождите 15 секунд.");

Thread.Sleep(1500);

}

else if (y == 1)

{

x = 0;

y++;

MessageBox.Show("Был введен неверный логин/пароль 3 раза,

подождите 20 секунд.");

Thread.Sleep(2000);

}

else

{

MessageBox.Show("Вы слишком много раз ввели неправильно

логин и(или) пароль");

Environment.Exit(0);

}

}

}

catch (Exception ex)

{

MessageBox.Show("Ошибка входа. "

+ ex.Message);

}

}

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

{

if (checkBoxPass.Checked)

textBoxPass.PasswordChar = '\0';

else

textBoxPass.PasswordChar = '\*';

}

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

{

Environment.Exit(0);

}

}

}

# **ПРИЛОЖЕНИЕ 2. Исходный программный код модуля «Создание пользователя»**

using OfficeLogistics.officelogisticsDataSetTableAdapters;

namespace OfficeLogistics

{

public partial class CreateUser : Form

{

public CreateUser()

{

InitializeComponent();

}

/// Сверка паролей с перенаправлением на метод

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

{

if (password.Text == confirmpass.Text)

{

SignUp(login.Text, password.Text, role.Text);

}

else

{

MessageBox.Show("Пароли не совпадают");

}

}

/// Метод регистрации пользователя

public void SignUp(string login, string password, string srole)

{

string log, pass;

int role = 0;

if (this.role.Text == "Администратор")

{

role = 1;

}

else

{

if (this.role.Text == "Менеджер")

{

role = 2;

}

else

{

MessageBox.Show("Роль не была выбрана");

}

}

log = login;

pass = password;

if (String.IsNullOrEmpty(log) || String.IsNullOrEmpty(pass))

{

MessageBox.Show("Не все поля заполнены");

return;

}

int pos;

if (pass.Length < 6) // длина пароля

{

MessageBox.Show("Password is too short");

return;

}

for (int i = 0; i < pass.Length; i++) // проверка на латиницу

{

if (!(Char.IsDigit(pass[i]) || pass[i] >= 'a' && pass[i] <= 'z' ||

pass[i] >= 'A' && pass[i] <= 'Z'))

{

MessageBox.Show("Пароль должен содержать только латинские символы");

return;

}

}

int countNum = 0;

for (int c = 0; c < pass.Length; c++)

{

if (pass[c] >= '0' && pass[c] <= '9')

{

countNum++;

}

}

if (countNum == 0)

{

MessageBox.Show("В пароле отсутствует цифра");

}

officelogisticsDataSet.UsersDataTable dataUsers = usersTableAdapter1.GetData();

try

{

usersTableAdapter1.Insert(dataUsers.Count() + 1, login, password, role);

MessageBox.Show("Пользователь добавлен");

}

catch

{

MessageBox.Show("Проверьте верность введённых данных");

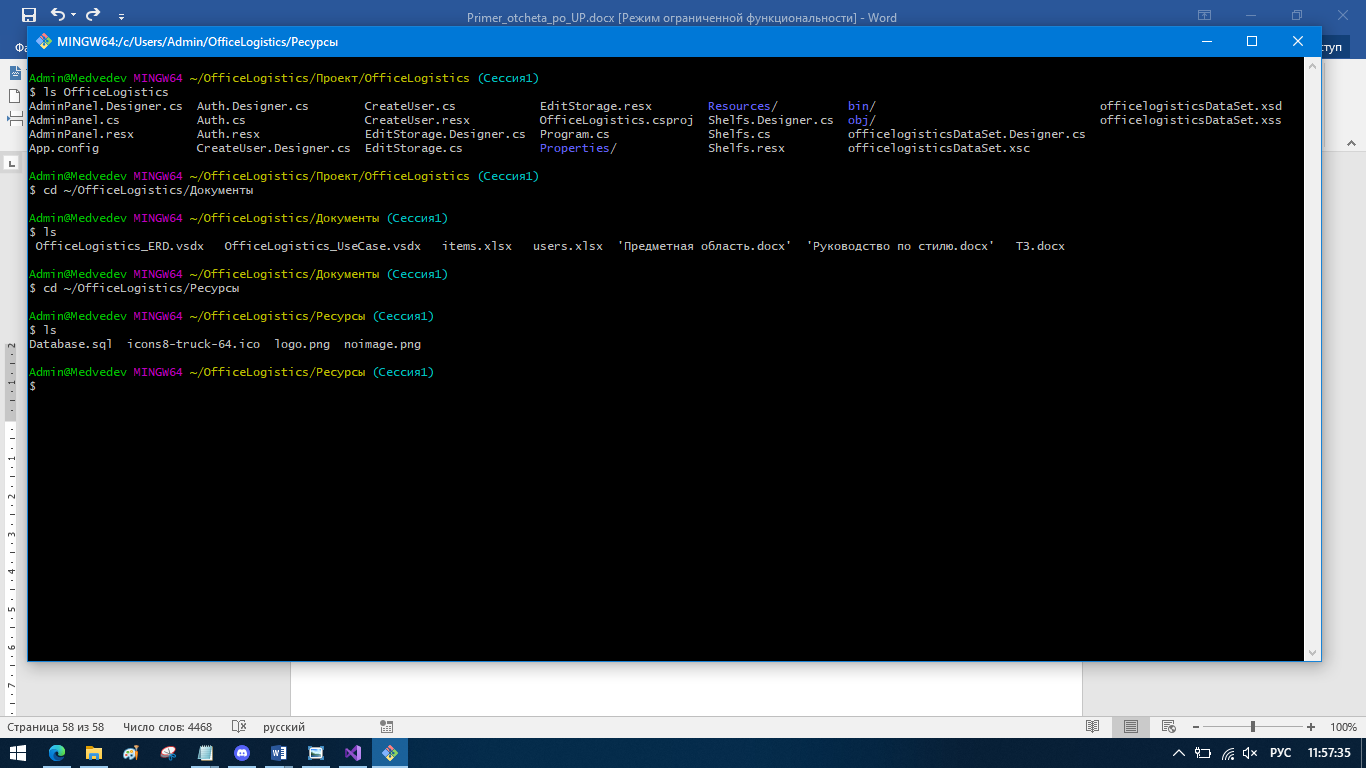
}

}

}

}

# **ПРИЛОЖЕНИЕ 3. Структура репозитория**



# **ПРИЛОЖЕНИЕ 4. Структура проекта**

