Краевое государственное автономное

профессиональное образовательное учреждение

«Ачинский техникум нефти и газа»

|  |  |
| --- | --- |
| Специальность | 09.02.04 Информационные системы (по отраслям) |
| Группа | ИС – 18 |

КУРСОВАЯ РАБОТА

ПМ.02 Участие в разработке информационных систем

|  |  |
| --- | --- |
| Тема: | Разработка клиент-серверной ИС «Оформление банковских карт» |

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | А.А. Назаров |
|  |  |  |
|  |  |  |
| Проверила: |  | О.В. Степанова |

Ачинск 2022

**Министерство образования Красноярского края**

**Краевое государственное АВТОНОМНОЕ профессиональное образовательное учреждение**

**«Ачинский техникум нефти и газа»**

ЗАДАНИЕ

на выполнение курсовой работы

|  |  |  |
| --- | --- | --- |
| Специальность\Профессия |  | 09.02.04 Информационные системы (по отраслям) |
| Профессиональный модуль\Дисциплина |  | ПМ.01 Эксплуатация и модификация информационных систем |

|  |  |  |
| --- | --- | --- |
| Рассмотрено  предметно-цикловой комиссией информатики и вычислительной техники  протокол № \_\_от «\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2021 г.  Председатель \_\_\_\_\_\_\_\_\_\_\_О.В. Степанова |  | «УТВЕРЖДАЮ»  зам. директора по УПР  \_\_\_\_\_\_\_\_\_\_\_ О.Н. Питенина  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2021 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Тема: | Модификация информационной системы «Оформление банковских карт» | | | | | | |
| Содержание: | ВВЕДЕНИЕ | | | | | | |
|  | ГЛАВА 1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА МОДИФИКАЦИЮ ИНФОРМАЦИОННОЙ СИСТЕМЫ | | | | | | |
|  | * 1. Основания для разработки | | | | | | |
|  | * 1. Разработка технического задания | | | | | | |
|  | ГЛАВА 2. РАЗРАБОТКА ЭСКИЗНОГО ПРОЕКТА | | | | | | |
|  | 2.1 Построение функциональной модели | | | | | | |
|  | * 1. Проектирование ER-модели | | | | | | |
|  | 2.3 Конструирование прототипа | | | | | | |
|  | ГЛАВА 3. РАЗРАБОТКА ТЕХНИЧЕСКОГО ПРОЕКТА | | | | | | |
|  | 3.1 Обоснование выбора средств разработки | | | | | | |
|  | 3.2 Описание основных программных модулей | | | | | | |
|  | ГЛАВА 4.ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ СИСТЕМЫ | | | | | | |
|  | ГЛАВА 5. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ | | | | | | |
|  | ЗАКЛЮЧЕНИЕ | | | | | | |
|  | Приложение | | | | | | |
| Дата выдачи задания: | |  | | Срок сдачи: | |  | |
| Руководитель работы | | |  | | О.В. Степанова | |
| Студент | | |  | | А.А. Назаров | |

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc75187487)

[ГЛАВА 1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА МОДИФИКАЦИЮ ИНФОРМАЦИОННОЙ СИСТЕМЫ 6](#_Toc75187488)

[1.1. ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ 6](#_Toc75187489)

[1.2. РАЗРАБОТКА ТЕХНИЧЕСКОГО ЗАДАНИЯ 7](#_Toc75187490)

[ГЛАВА 2. РАЗРАБОТКА ЭСКИЗНОГО ПРОЕКТА 11](#_Toc75187491)

[2.1. ПОСТРОЕНИЕ ФУНКЦИОНАЛЬНОЙ МОДЕЛИ 11](#_Toc75187492)

[2.2. ПРОЕКТИРОВАНИЕ ER-МОДЕЛИ 11](#_Toc75187493)

[2.3. КОНСТРУИРОВАНИЕ ПРОТОТИПА 18](#_Toc75187494)

[ГЛАВА 3. РАЗРАБОТКА ТЕХНИЧЕСКОГО ПРОЕКТА 28](#_Toc75187495)

[3.1. ОБОСНОВАНИЕ ВЫБОРА СРЕДСТВ РАЗРАБОТКИ 28](#_Toc75187496)

[3.2 ОПИСАНИЕ ОСНОВНЫХ ПРОГРАММНЫХ МОДУЛЕЙ 29](#_Toc75187497)

[ГЛАВА 4. ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЙ СИСТЕМ 51](#_Toc75187498)

[ГЛАВА 5. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 55](#_Toc75187499)

[ЗАКЛЮЧЕНИЕ 57](#_Toc75187500)

[СПИСОК ИСПОЛЬЗУВАЕМЫХ ИСТОЧНИКОВ 59](#_Toc75187501)

Изм.

Лист

№ докум.

Подп.

Дата

Лист

3

КР. 09.02.04. ОО. 6664. ПЗ

Разраб.

Назаров А.А,.

Провер.

Степанова О.В.

Реценз.

Н. Контр.

Степанова О.В.

Утв.

Модификация информационной системы «Косметическая фирма»

Лит.

Листов

64

АТНГ гр. ИС -18

# ВВЕДЕНИЕ

Бурное развитие электронных технологий во всех сферах жизни общества приводит к тому, что они все активнее проникают в денежное обращение. В настоящих условиях уровень развития информационных технологий в банковской сфере объективно определяет появление новых способов и механизмов проведения процедур платежа. В настоящее время наблюдается высокая активность российских кредитных организации в развитии системы электронных платежей, сокращение бумажной технологий. Для этого процесса характерно: увеличение эмиссии пластиковых карт, рост оборотов и остатков по карточным счетам, расширение спектра услуг с использованием банковских карт. Для любого отделения банка важно иметь программу, с помощью которой можно управлять информацией, касающейся банковских карт.

Информационные системы используются почти во всех современных отделениях банков.

Основанием разработки послужил ряд причин:

* отсутствие АИС для учета банковских карт;
* возможность случайной ошибки в ручном подсчете;
* невозможность быстро справиться с большим объемом информации;
* потеря данных;
* отсутствие возможности обновления всех данных.

Целью данной курсовой работы является модификация информационной системы, которая автоматизирует оформление банковских карт.

Задачи:

1. Определить назначение разработки системного проекта;
2. Определить требования к функциональным характеристикам;
3. Определить состав выполняемых функций;
4. Определить входные и выходные данные;
5. Определить требования к составу технических средств;
6. Определить требования к информационной и программной совместимости;
7. Построить функциональную модель;
8. Осуществить проектирование ER-модели данных (или описание алгоритма решения задачи);
9. Редактирование таблицы базы данных SQL Server в среде VisualStudio с написанием программного кода;
10. Протестировать базу данных и приложения системы;
11. Написать руководство пользователя.

# ГЛАВА 1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ ПРОГРАММНОГО ПРОДУКТА

* 1. ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ

Перед тем, как приступить к разработке проекта, следует проанализировать существующие на данный момент системы подобного характера и тематики.

С помощью этого мы можем сделать верные выводы о том, как необходимо создавать собственную систему на основе достоинств и недостатков рассмотренных ресурсов.

В качестве первого аналога был выбран сайт rncb.ru (рисунок 1) (<https://www.rncb.ru/> ).

Достоинства:

* Вход через Госуслуги;
* Самообслуживание.

Недостатки:

* Много абсолютно ненужной информации;
* Сайт зависает на слабых компьютерах;
* Сложно разобраться в оформлении карты без специалиста.

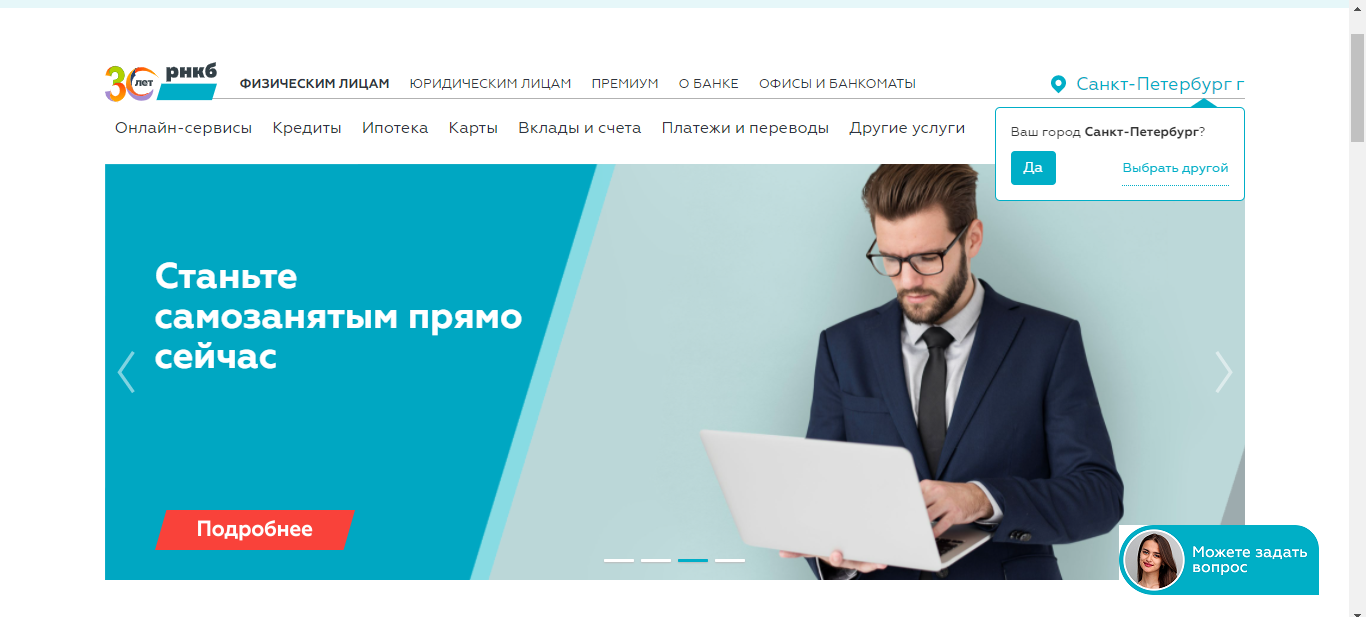


Рисунок 1 - rncb.ru

В качестве второго аналога был выбран сайт gazprombank.ru (рисунок 2) (<https://www.gazprombank.ru/>).

Достоинства:

* Удобный и приятный интерфейс;
* Самообслуживание.

Недостатки:

* Отсутствует доступ к контенту при слабом интернет-соединении;
* Возникают ошибки в ходе оформления карты;
* Не приходит код подтверждения.

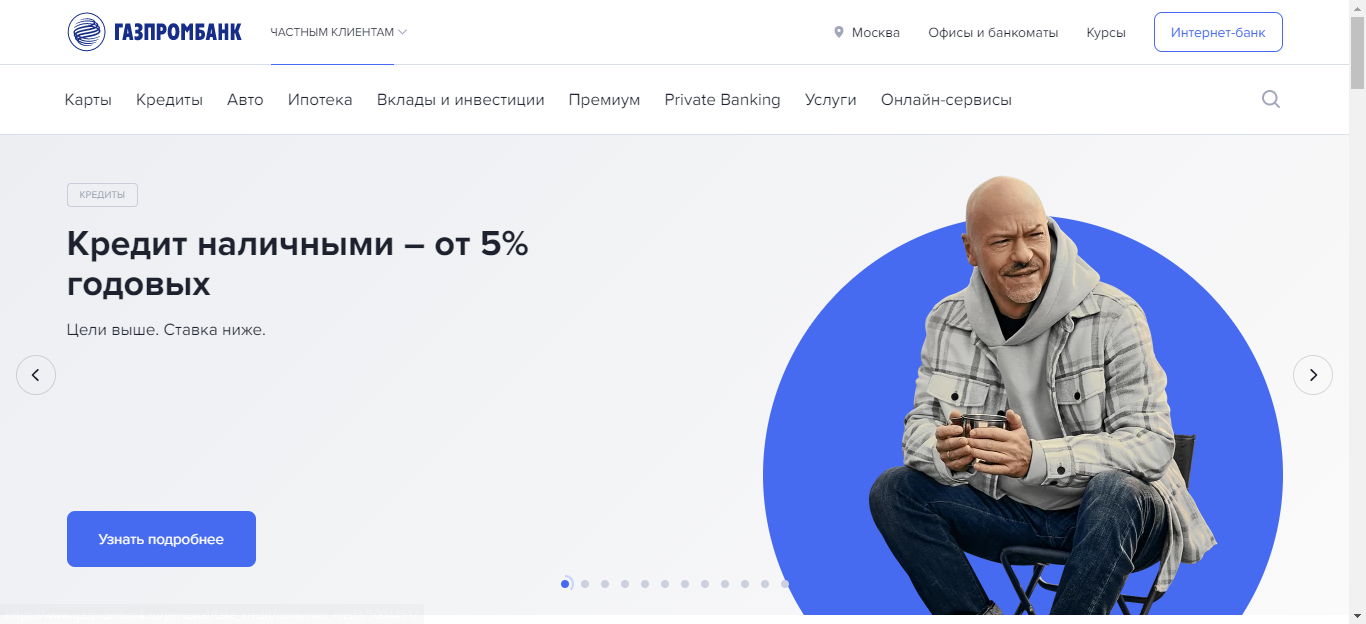


Рисунок 2 - gazprombank.ru

На основании исследований, можно сформулировать следующие выводы:

1. В настоящее время любая организация может улучшить качество своей работы, пользуясь современными автоматизированными ИС;
2. Использование таких систем явно лидирует по сравнению с бумажной волокитой, здесь можно подчеркнуть такие плюсы, как скорость, безопасность, упорядоченность и понятность.
   1. РАЗРАБОТКА ТЕХНИЧЕСКОГО ЗАДАНИЯ

*Общие сведения:*

* Полное наименование: информационная база для оформления банковских карт;
* Наименование предприятий разработчика и заказчика системы и их реквизиты:

Заказчик: Боров С.Д.

Разработчик: Назаров А.А.

Отделение банка «Главбанк» (ООО «Главбанк»);

* Плановые сроки начала и окончания работы по созданию системы: 07.04.2021-28.06.2021;
* Порядок оформления и предъявления заказчику результатов работ по созданию системы, по изготовлению и наладке отдельных средств и программно-технических комплексов системы: согласно стандарту, ГОСТ 34.602-89.

Разработанный программный продукт представляет собой программу, которая объединяет все сведения необходимые для систематизации и упорядочивания процесса оформления банковских карт. Разработанная программа обладает следующей функциональностью:

*Назначения:*

* Хранение данных о клиентах, картах, договорах, сотрудниках, заявках (на оформление и изготовление);
* Работа со списками клиентов (и всеми связующими), заявок, сотрудников (возможность добавления, редактирования, обновление и удаления данных);
* Регистрация банковской карты.

*Цель:*

* Совершенствование ведения учёта карт, клиентов и заявок;
* Эффективная обработка заявок;
* Снижение рисков утери данных о клиентах, заявках, картах и т.д.

Цель банка «Главбанк» заключается:

* Извлечение прибыли всеми доступными законными способами;
* Наиболее качественное и полное удовлетворение потребностей клиентов;
* Иметь постоянно обновлённую информацию о клиентах и картах;
* Обеспечить оперативное, максимально удобное и простое, для пользователей, предоставление информации.

*Требование системы в целом:*

* Уменьшение времени по учёту данных о клиентах и картах;
* Технические средства должны быть установлены так, чтобы обеспечивались их безопасная эксплуатация и техническое обслуживание.

*Требования к надежности и безопасности:*

* Конфиденциальность - обеспечение доступа к информации только авторизованным пользователям.
* Целостность - обеспечение достоверности и полноты информации и методов ее обработки.
* Доступность - обеспечение доступа к информации и связанным с ней активам авторизованных пользователей по мере необходимости.

*Требования к составу аппаратных и программных средств:*

Для работы с программным обеспечением отделением банка «Главбанк» необходимым прикладным программным обеспечением является SQL Server.

*Техническое обеспечение ИС-компьютер:*

* Windows 7 и выше.

Для установки SQL Server минимум необходимо:

* Модель процессора - Core i5 7400;
* Количество ядер процессора – 4;
* Процессор с частотой не ниже 1 ГГц;
* Автоматическое увеличение частоты – 3500 МГц;
* Размер оперативной памяти не менее 512 МБ;
* Свободное место на жестком диске 4,2 ГБ.

Данная ИС разрабатывается для автоматизации работы сотрудников отделения банка «Главбанк».

При работе системы, сотрудник должен решать следующие задачи:

* Получать доступ ко всем данным, в которых должна содержаться вся необходимая информация;
* Просматривать данные, при необходимости, редактировать их (добавлять или удалять).
* Управление списков клиентов, сотрудников, карт и заявок.

*Стадии разработки:*

* Разработка технического задания;
* Рабочее проектирование;
* Внедрение.

# ГЛАВА 2. РАЗРАБОТКА ЭСКИЗНОГО ПРОЕКТА

2.1. ПОСТРОЕНИЕ ФУНКЦИОНАЛЬНОЙ МОДЕЛИ

Разработку программного обеспечения начнем с анализа требований к будущему программному продукту. Построим общую модель предметной области как некоторой части реального мира, с которой будет тем или иным способом взаимодействовать разрабатываемое программное обеспечение.

Таким образом, определим, как разрабатываемая система будет взаимодействовать с приемниками и источниками информации.

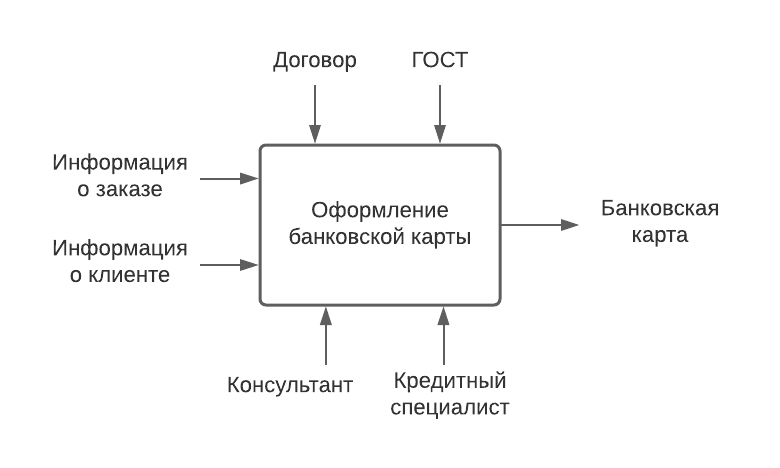


Рисунок 3-Функциональная модель оформления банковской карты

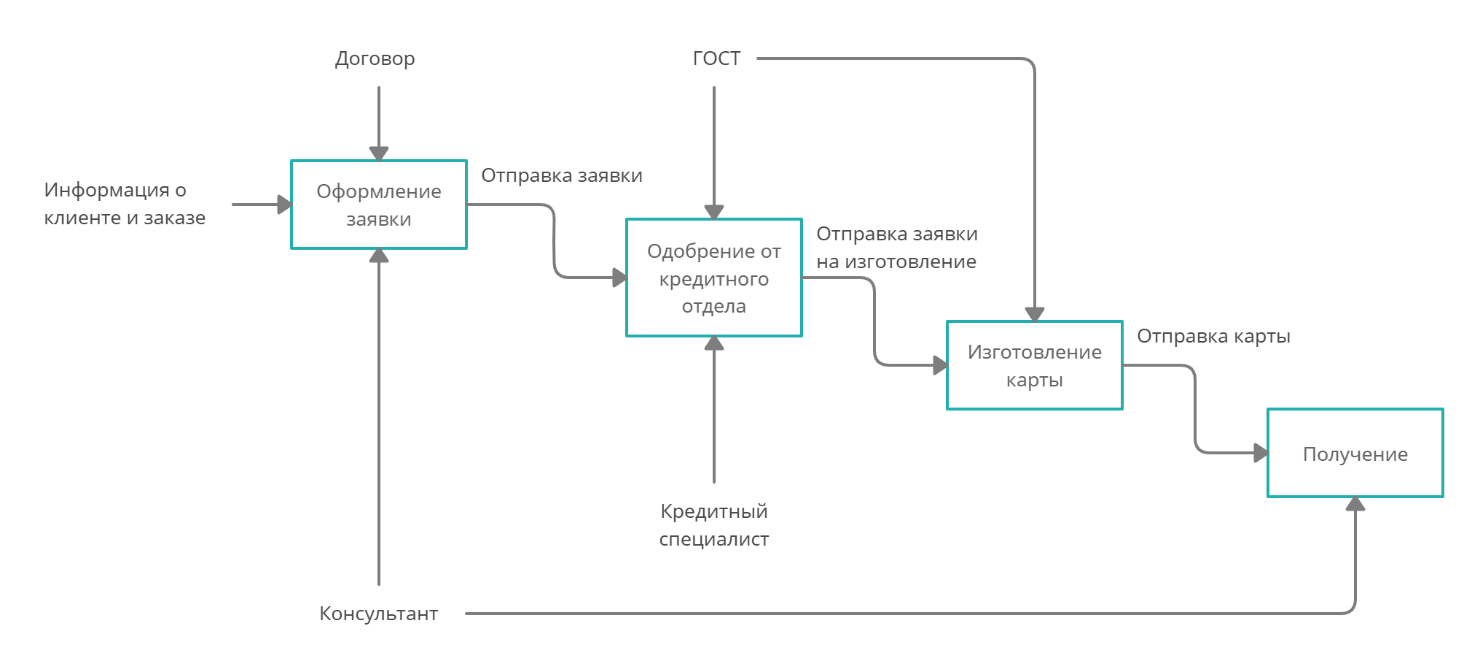


Рисунок 4 - Детальная модель оформления банковской карты

## 2.2. ПРОЕКТИРОВАНИЕ ER**-**МОДЕЛИ

Теперь разработаем диаграмму «сущность – связь» (ER – модель данных) которая обеспечивает способ определения данных и отношений между ними. Модель данных включает сущности и связи между ними.

Сущность – это класс однотипных реальных или абстрактных объектов (людей, событий, состояний, предметов и т.п.), информация о которых имеет существенное значение для рассматриваемой предметной области.

Диаграммы «сущность - связь» определяют спецификации структур данных программного обеспечения. В модели данных, разрабатываемой автоматизированной системы оформления банковских карт можно выделить следующие сущности:

1. Сущность «Сотрудники»;
2. Сущность «Счет»;
3. Сущность «Карта»;
4. Сущность «Заявка»;
5. Сущность «Клиент»;
6. Сущность «Договор».

Проанализируем связи между сущностями, чтобы определить, можно ли уменьшить число подсхем при помощи минимизации данных.

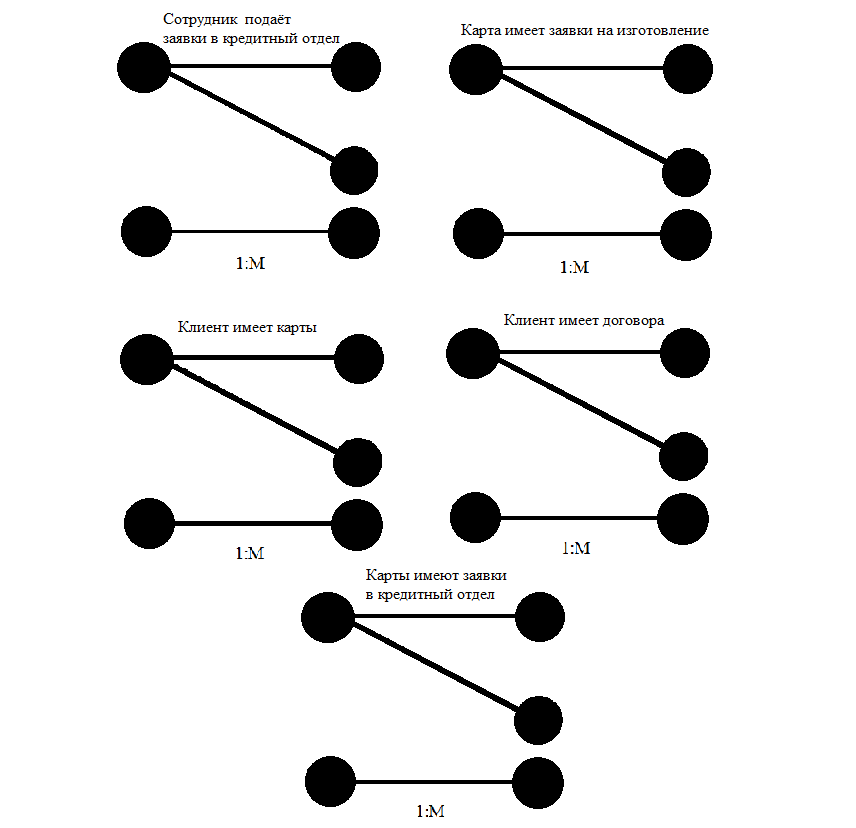


Рисунок 5 - Точечная диаграмма

Основные отношения между указанными сущностями показаны на рисунке 5.

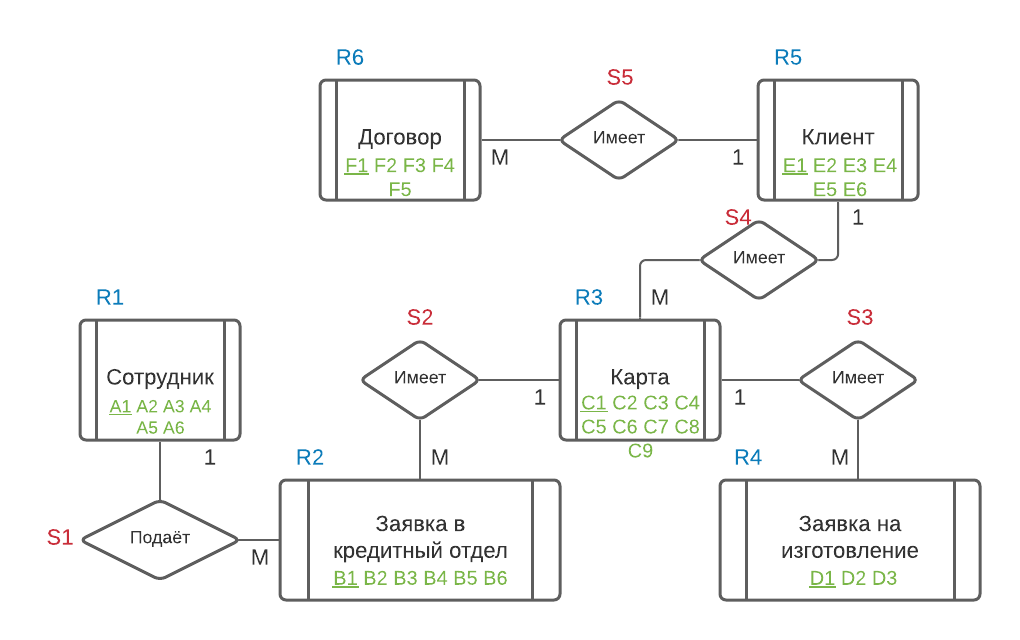


Рисунок 6 - ER диаграмма

Между сущностями «Сотрудник» и «Заявка в кредитный отдел», «Карта» и «Заявка в кредитный отдел», «Карта» и «Заявка на изготовление» , «Клиент» и «Карта», «Клиент» и «Договор» тип связей 1:М и класс принадлежности со стороны многих обязательный, значит, мы можем провести минимизацию в обоих случаях.

Построим схему базы данных, используя общий прием:

Q0= {R1, R2, R3, R4, R5, R6, S1, S2, S3, S4, S5}, где:

R1= {A1, A2, A3, A4, A5, A6}

R2= {B1, B2, B3, B4, B5, B6}

R3= {C1, C2, C3, C4, C5, C6, C7, C8, C9}

R4= {D1, D2, D3}

R5= {E1, E2, E3, E4, E5, E6}

R6= {F1, F2, F3, F4, F5}

S1= {A1, B6}

S2= {B1, C2}

S3= {C1, D2}

S4= {E1, C7}

S5= {E1, F2}

Рассмотрим возможность минимизации схемы базы данных:

Минимизация:

Q1= {R1, R2\*, R3, R4, R5, R6, S2, S3, S4, S5}, где:

R1= {A1, A2, A3, A4, A5, A6}

R2\*= {R2 ∪ S1} = {B1, B2, B3, B4, B5, B6, B6}

R3= {C1, C2, C3, C4, C5, C6, C7, C8, C9}

R4= {D1, D2, D3}

R5= {E1, E2, E3, E4, E5, E6}

R6= {F1, F2, F3, F4, F5}

S2= {B1, C2}

S3= {C1, D2}

S4= {E1, C7}

S5= {E1, F2}

Q2= {R1, R2\*, R3\*, R4, R5, R6, S3, S4, S5}, где:

R1= {A1, A2, A3, A4, A5, A6}

R2\*= {R2 ∪ S1} = {B1, B2, B3, B4, B5, B6, B6}

R3\*= {R3 ∪ S2} = {C1, C2, C3, C4, C5, C6, C7, C8, C9, B1}

R4= {D1, D2, D3}

R5= {E1, E2, E3, E4, E5, E6}

R6= {F1, F2, F3, F4, F5}

S3= {C1, D2}

S4= {E1, C7}

S5= {E1, F2}

Q3= {R1, R2\*, R3\*\*, R4, R5, R6, S4, S5}, где:

R1= {A1, A2, A3, A4, A5, A6}

R2\*= {R2 ∪ S1} = {B1, B2, B3, B4, B5, B6, B6}

R3\*\*= {R3\* ∪ S3} = {C1, C2, C3, C4, C5, C6, C7, C8, C9, B1, D2}

R4= {D1, D2, D3}

R5= {E1, E2, E3, E4, E5, E6}

R6= {F1, F2, F3, F4, F5}

S4= {E1, C7}

S5= {E1, F2}

Q4= {R1, R2\*, R3\*\*\*, R4, R5, R6, S5}, где:

R1= {A1, A2, A3, A4, A5, A6}

R2\*= {R2 ∪ S1} = {B1, B2, B3, B4, B5, B6, B6}

R3\*\*\*= {R3\*\* ∪ S4} = {C1, C2, C3, C4, C5, C6, C7, C8, C9, B1, D2, E1}

R4= {D1, D2, D3}

R5= {E1, E2, E3, E4, E5, E6}

R6= {F1, F2, F3, F4, F5}

S5= {E1, F2}

Q5= {R1, R2\*, R3\*\*\*, R4, R5\*, R6}, где:

R1= {A1, A2, A3, A4, A5, A6}

R2\*= {R2 ∪ S1} = {B1, B2, B3, B4, B5, B6, B6}

R3\*\*\*= {R3\*\* ∪ S4} = {C1, C2, C3, C4, C5, C6, C7, C8, C9, B1, D2, E1}

R4= {D1, D2, D3}

R5\*= {R5 ∪ S5} = {E1, E2, E3, E4, E5, E6, F2}

R6= {F1, F2, F3, F4, F5}

Минимизация прошла успешно.

На следующем шаге определяем атрибуты каждой сущности и уточняем их типы.

Таблица 1. Сущность «Сотрудник»

|  |  |  |
| --- | --- | --- |
| Атрибуты | Семантика | Домен |
| А1 | Код сотрудника | Integer |
| А2 | Логин | Varchar(max) |
| A3 | Пароль | Varchar(max) |
| A4 | ФИО | Varchar(max) |
| A5 | Роль | Integer |
| A6 | Обработано | Integer |

Таблица 2. Сущность «Заявка в кредитный отдел»

|  |  |  |
| --- | --- | --- |
| Атрибуты | Семантика | Домен |
| В1 | Код заявки | Integer |
| В2 | Номер карты | Varchar(16) |
| В3 | Информация о займе | Varchar(max) |
| В4 | Сумма займа | Integer |
| В5 | Срок | Integer |
| B6 | Код сотрудника | Integer |

Таблица 3. Сущность «Карта»

|  |  |  |
| --- | --- | --- |
| Атрибуты | Семантика | Домен |
| С1 | Номер карты | Integer |
| С2 | Деньги | Integer |
| C3 | Тип карты | Integer |
| C4 | Действие карты | Varchar(10) |
| C5 | CVV | Integer |
| C6 | PIN-код | Integer |
| C7 | Код клиента | Integer |
| C8 | Одобрение | Bit |
| C9 | Изготовлена | Integer |

Таблица 4. Сущность «Заявка на изготовление»

|  |  |  |
| --- | --- | --- |
| Атрибуты | Семантика | Домен |
| D1 | Код заявки | Integer |
| D2 | Номер карты | Varchar(16) |
| D3 | Дата подачи | Datetime |

Таблица 5. Сущность «Клиент»

|  |  |  |
| --- | --- | --- |
| Атрибуты | Семантика | Домен |
| E1 | Код клиента | Integer |
| E2 | ФИО | Varchar(max) |
| E3 | Номер паспорта | Varchar(15) |

Продолжение таблицы 5.

|  |  |  |
| --- | --- | --- |
| Атрибуты | Семантика | Домен |
| E4 | Кем выдан паспорт | Varchar(max) |
| E5 | Номер телефона | Varchar(50) |
| E6 | Дата рождения | Date |

Таблица 6. Сущность «Договор»

|  |  |  |
| --- | --- | --- |
| Атрибуты | Семантика | Домен |
| F1 | Код договора | Integer |
| F2 | Код клиента | Integer |
| F3 | Дата заключения | Datetime |
| F4 | Сумма | Integer |
| F5 | Срок | Integer |

Проверим работоспособность модели, построив схему данных в Microsoft SQL Server:

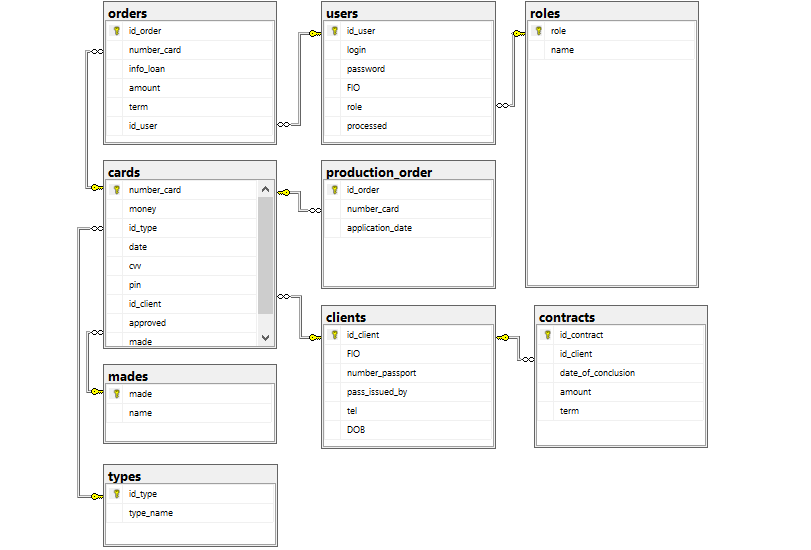


Рисунок 7 – Схема данных

## 2.3. КОНСТРУИРОВАНИЕ ПРОТОТИПА

Должна быть разработана форма авторизации для сотрудников (в соответствии с рисунком 8).

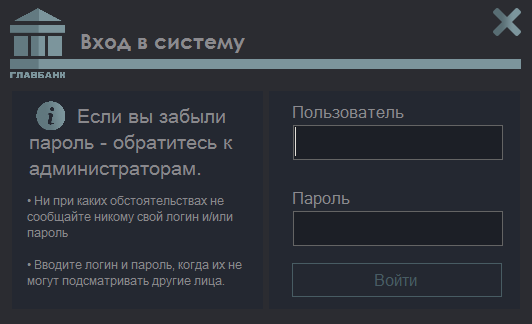


Рисунок 8 - форма авторизации пользователей

Форма авторизации содержит поля для ввода Логина и Пароля, а также кнопку «Войти».

Введем логин и пароль администратора на форме авторизации и перейдем в панель (в соответствии с рисунком 9).

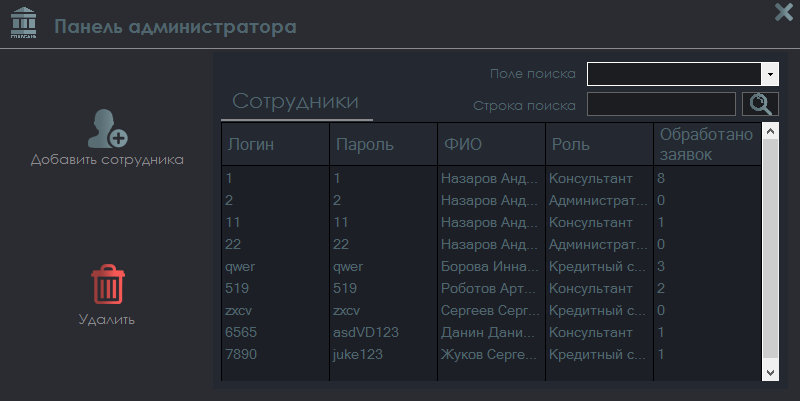


Рисунок 9 - Панель администратора

На этой форме администратор имеет доступ к сотрудникам, где можно добавлять и удалять информацию о них.

Нажав на кнопку «Добавить сотрудника», перейдём на форму для регистрации сотрудника (в соответствии с рисунком 10).

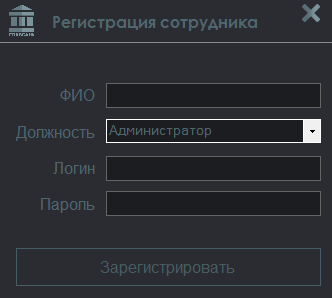


Рисунок 10 - Форма регистрации сотрудника

На этой форме расположены поля для регистрации нового пользователя, а также, кнопка «Зарегистрировать».

Вернемся на предыдущую форму, где присутствует фильтр при помощи которого мы можем находить нужную запись по введённой информации (в соответствии с рисунком 11).

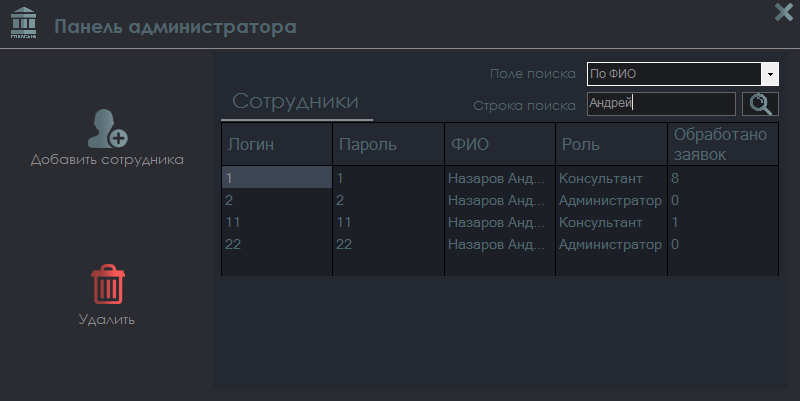


Рисунок 11 - поиск через Фильтр

Также на панели администратора есть кнопка удаления, которая удаляет выбранную запись.

Вернемся на форму регистрации, введем логин и пароль консультанта и нажмем кнопку «Войти» для входа в панель сотрудника (в соответствии с рисунком 12).

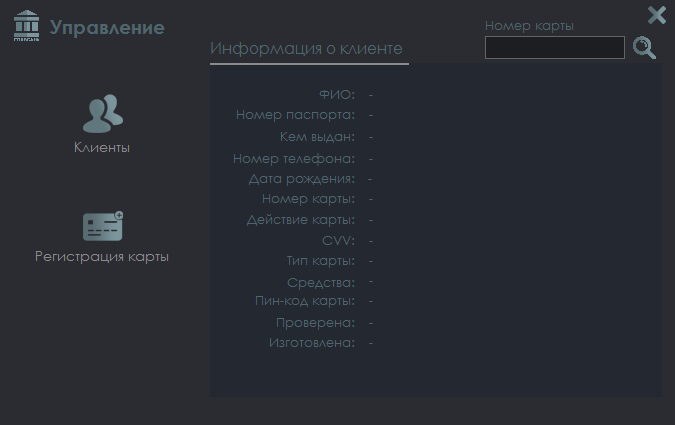


Рисунок 12 – Панель сотрудника

На этой форме консультант имеет доступ к информации об определённом клиенте. Также на форме присутствуют кнопки регистрации карты «Регистрация карты» и кнопка просмотра всех клиентов отделения «Клиенты».

Введем номер карты определённого клиента в поле «Номер карты» и нажмем кнопку поиска для моментального получения информации (в соответствии с рисунком 13).

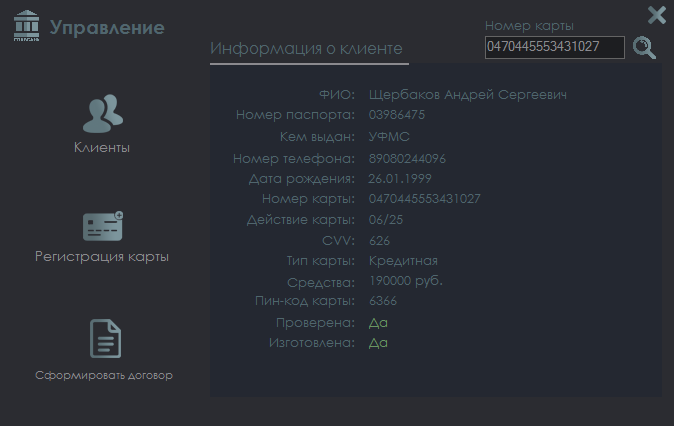


Рисунок 13 – Информация о клиенте

После появления информации рядом также появляется кнопка «Сформировать договор», при нажатии на которую формируется договор в формате Microsoft Word и сохраняется на рабочий стол.

При нажатии на кнопку «Регистрация карты» появляется форма регистрации карты (в соответствии с рисунком 14).

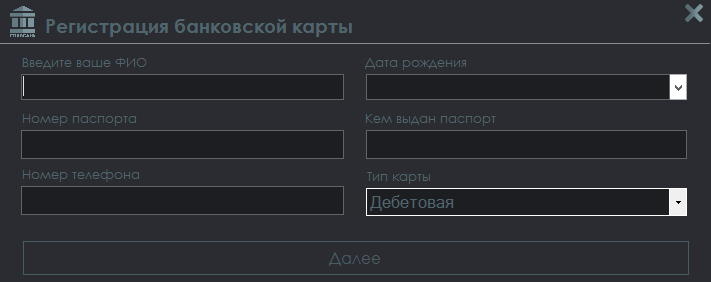


Рисунок 14 – Форма регистрации карты

После заполнения всех полей на форме нужно нажать кнопку «Далее», чтобы продолжить регистрацию дебетовой карты, либо выбрать другой тип карты (дебетовая, кредитная, рассрочная) в поле «Тип карты», после чего откроется дополнительные поля для оформления кредитной или рассрочной карты (в соответствии с рисунком 15).

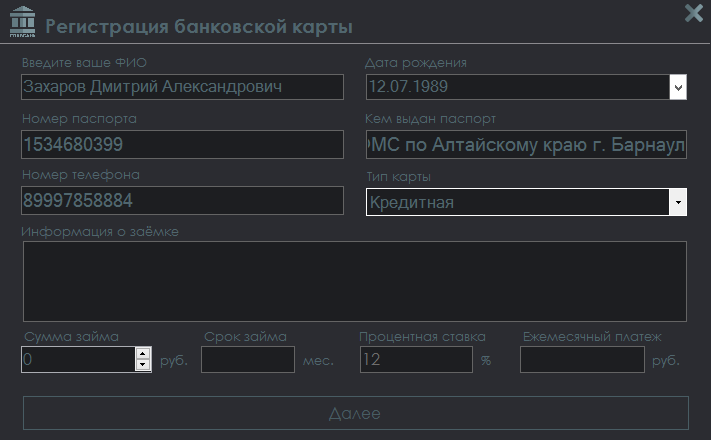


Рисунок 15 – Продолжение регистрации карты

После заполнения данных о карте нужно нажать далее для продолжения (в соответствии с рисунком 16).

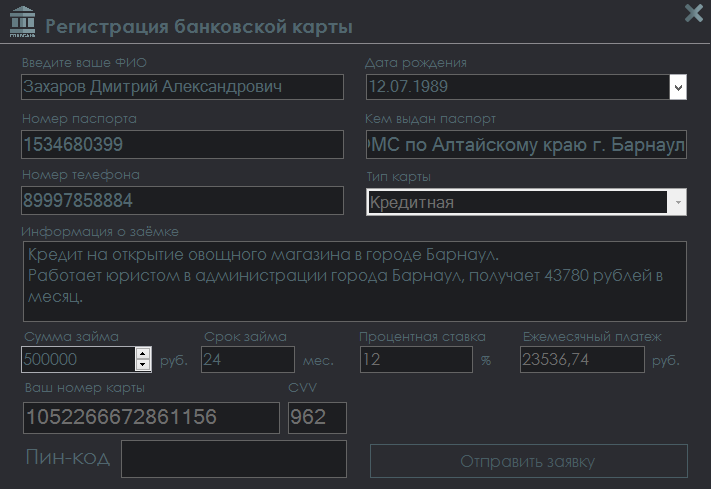


Рисунок 16 – Последний этап регистрации карты

Далее нужно ввести пин-код, который придумает клиент, и отправить заявку в кредитный отдел нажав на кнопку «Отправить заявку». Форма закрывается, и мы возвращаемся на главную форму консультанта (рис. 12).

При нажатии на кнопку «Клиенты» открывается форма со всеми клиентами отделения (в соответствии с рисунком 17).

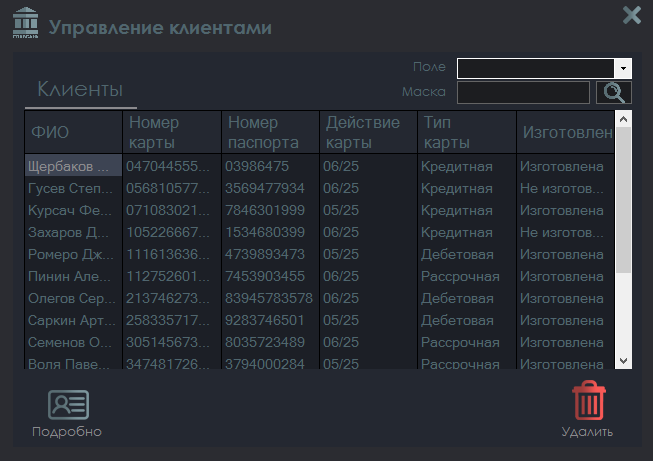


Рисунок 17 – Клиенты

На этой форме представлены все клиенты отделения. Нажав на кнопку «Удалить», можно удалить клиента из таблицы, предварительно выбрав его. Присутствует поиск, с помощью которого можно найти нужного клиента.

При нажатии на кнопку «Отследить», которая есть не у всех клиентов, открывается форма, где можно просмотреть весь процесс изготовление карты (в соответствии с рисунком 18).

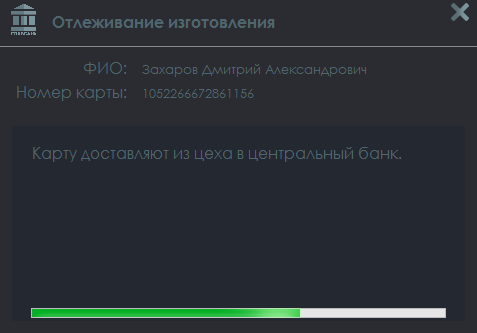


Рисунок 18 – Форма отслеживания изготовления

Перейдём обратно на форму с клиентами (рис. 17), выберем любого клиента и нажмем на кнопку «Подробно». Откроется главная панель консультанта, где формируется подробная информация о данном клиенте (в соответствии с рисунком 13).

Вернемся на форму авторизации и введем логин и пароль кредитного специалиста. Откроется меню кредитного отдела (в соответствии с рисунком 19).

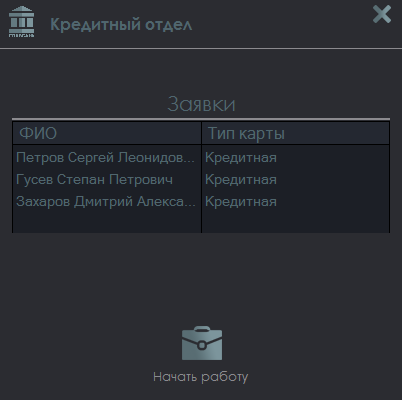


Рисунок 19 – Форма кредитного отдела

На данной форме представлены все заявки кредитного отдела и кнопка для начала работы с определенной заявкой. Выбрав нужную заявку и нажав на кнопку «Начать работу» откроется форма кредитного специалиста (в соответствии с рисунком 20).

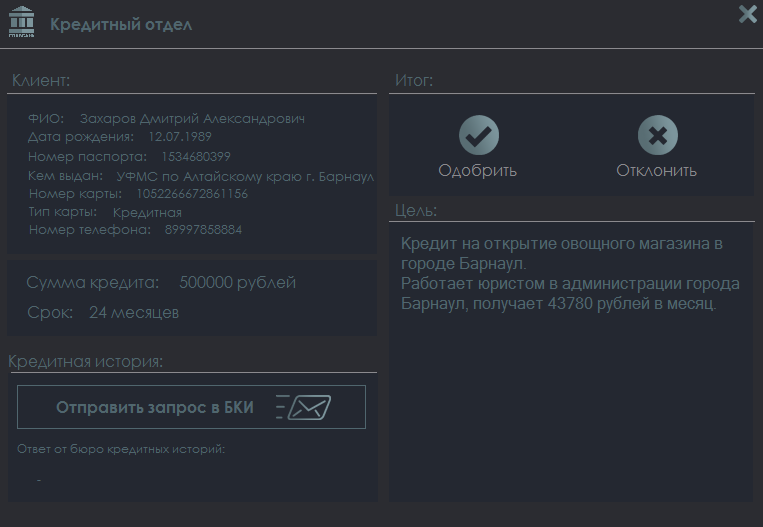


Рисунок 20 – Форма кредитного специалиста

На данной форме решается как поступить с клиентом, отталкиваясь на информацию, которая здесь находится. На форме присутствуют кнопки «Одобрить» и «Отклонить», при нажатии на которые выносится соответствующий вердикт о данном клиенте. Также на форме есть кнопка для отправки запроса в БКИ (бюро кредитных историй) и получения ответа для точного решения о кредите.

# ГЛАВА 3. РАЗРАБОТКА ТЕХНИЧЕСКОГО ПРОЕКТА

## 3.1. ОБОСНОВАНИЕ ВЫБОРА СРЕДСТВ РАЗРАБОТКИ

Язык C# как средство разработки и реализации программ обладает рядом несомненных, достоинств. Он хорошо организован, строг, большинство его конструкций логичны и удобны.

Мощная библиотека классов платформы .NET берет на себя массу рутинных операций, что дает возможность решать более сложные задачи, используя готовые «Строительные блоки». Все это позволяет расценивать C# как перспективную замену языков Паскаль, BASIC и C++ при обучении программированию.

Немаловажно, что C# является не учебным, а профессиональным языком, предназначенным для решения широкого спектра задач, и в первую очередь – в быстро развивающейся области создания распределённых приложений.

Многие разработчики хотели бы использовать современный язык, который позволял бы писать, читать и сопровождать программы с простотой Visual Basic и в то же время давал мощь и гибкость C++, обеспечивал доступ ко всем функциональным возможностям системы, взаимодействовал бы с существующими программами и легко работал с возникающими Web стандартами.

Учитывая все подобные пожелания, Microsoft разработала новый язык - C#. В него входит много полезных особенностей - простота, объектная ориентированность, типовая защищенность, поддержка совместимости версий и многое другое. Данные возможности позволяют быстро и легко разрабатывать приложения. При создании C#, его авторы учитывали достижения многих других языков программирования: C++, C, Java, SmallTalk, Delphi, Visual Basic и т.д.

По причине того, что C# разрабатывался с чистого листа, у его авторов была возможность, оставить в прошлом все неудобные и неприятные особенности, любого из предшествующих ему языков. В результате получился действительно простой, удобный и современный язык, по мощности не уступающий С++, но существенно повышающий продуктивность разработок.

Microsoft SQL Server — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

QL Server — это база данных, которая уже шесть лет признается наименее уязвимой по результатам тестов на уязвимость в Национальном институте стандартов и технологий США (NIST).

## 3.2 ОПИСАНИЕ ОСНОВНЫХ ПРОГРАММНЫХ МОДУЛЕЙ

Код на кнопку «Войти»:

private void button1\_Click(object sender, EventArgs e)

{

string SqlText = "SELECT \* FROM users ";

SqlDataAdapter da = new SqlDataAdapter(SqlText, ConnStr);

DataSet ds = new DataSet();

da.Fill(ds, "[users]");

DataTable dt = ds.Tables[0];

foreach (DataRow row in dt.Rows)

{

Form2 f2 = new Form2();

Form3 f3 = new Form3();

Form7 f7 = new Form7();

id\_user = row[0].ToString();

var login = row[1].ToString();

var password = row[2].ToString();

var role = row[4].ToString();

label4.Visible = false;

if ((login == textBox1.Text) && (password == textBox2.Text))

{

if (role == "1") f2.Show();

else if (role == "0") f3.Show();

else f7.Show();

return;

}

else label4.Visible = true;

}

}

При входе в учётную запись администратора нажимаем на кнопку «Добавить сотрудника»:

private void button1\_Click(object sender, EventArgs e)

{

Form6 f6 = new Form6();

f6.Show();

this.Hide();

}

Открывается окно регистрации. Код на кнопку «Зарегистрировать»:

private void button1\_Click(object sender, EventArgs e)

{

label6.Visible = false;

if (textBox1.Text != "" && textBox2.Text != "" && textBox3.Text != "")

{

Form3 f3 = new Form3();

bool there = false;

label4.Visible = false;

string get\_users = "SELECT \* FROM users ";

SqlDataAdapter da = new SqlDataAdapter(get\_users, Program.f1.ConnStr);

DataSet ds = new DataSet();

da.Fill(ds, "[clients]");

DataTable dt = ds.Tables[0];

foreach (DataRow row in dt.Rows)

{

var login = row[1].ToString();

if (login == textBox1.Text)

{

there = true;

label4.Visible = true;

return;

}

}

if (!there)

{

string addUser = "INSERT INTO users ([login], [password], [FIO], [role], [processed]) VALUES ('"

+ textBox2.Text + "', '"

+ textBox3.Text + "', '"

+ textBox1.Text + "', '"

+ comboBox1.SelectedIndex + "', '"

+ 0 + "')";

MyExecuteNonQuery(addUser);

this.Hide();

f3.Show();

}

}

else label6.Visible = true;

}

При нажатии на кнопку «Поиск» в фильтре:

private void button2\_Click(object sender, EventArgs e)

{

string CommandText = "SELECT " +

"[users].[id\_user], " +

"[users].[login] as [Логин], " +

"[users].[password] as [Пароль], " +

"[users].[FIO] as [ФИО], " +

"[roles].[name] as [Роль], " +

"[users].[processed] as [Обработано заявок] " +

"FROM users, roles " +

"WHERE " +

"([users].[role] = [roles].[role])";

if (textBox1.Text != "")

{

if (comboBox1.SelectedIndex == 0)

CommandText = CommandText + "AND ([users].[login] LIKE '%" + textBox1.Text + "%')";

if (comboBox1.SelectedIndex == 1)

CommandText = CommandText + "AND ([users].[password] LIKE '%" + textBox1.Text + "%') ";

if (comboBox1.SelectedIndex == 2)

CommandText = CommandText + "AND ([users].[FIO] LIKE '%" + textBox1.Text + "%') ";

if (comboBox1.SelectedIndex == 3)

CommandText = CommandText + "AND ([roles].[name] LIKE '%" + textBox1.Text + "%') ";

}

SqlDataAdapter da = new SqlDataAdapter(CommandText, Program.f1.ConnStr);

DataSet ds = new DataSet();

da.Fill(ds, "[clients]");

dataGridView1.DataSource = ds.Tables["[clients]"].DefaultView;

}

При нажатии на кнопку «Удалить»:

private void button3\_Click(object sender, EventArgs e)

{

int index = dataGridView1.CurrentRow.Index;

string DeleteUser = "DELETE FROM [users] WHERE ([users].[id\_user] = '" + Convert.ToString(dataGridView1[0, index].Value) + "')";

dataGridView1.Rows.RemoveAt(index);

MyExecuteNonQuery(DeleteUser);

}

При входе на учетную запись консультанта открывается форма. Код нажатия на кнопку «Получить информацию»:

public void button2\_Click(object sender, EventArgs e)

{

getInfo();

}

Функция getInfo():

public void getInfo()

{

string getInfoSql = "SELECT \* FROM [clients], [cards], [types] " +

"WHERE ([clients].[id\_client] = [cards].[id\_client]) " +

"AND ([types].[id\_type] = [cards].[id\_type])";

SqlDataAdapter da = new SqlDataAdapter(getInfoSql, Program.f1.ConnStr);

DataSet ds = new DataSet();

da.Fill(ds, "[cards]");

DataTable dt = ds.Tables[0];

if (textBox1.Text.Length >= 8)

{

label3.Visible = false;

button2.Top = 123;

button1.Top = 169;

button3.Top = 215;

}

else

{

label3.Visible = true;

button2.Top = 136;

button1.Top = 182;

button3.Top = 228;

}

foreach (DataRow row in dt.Rows)

{

Form2 f1 = new Form2();

Form3 f2 = new Form3();

if (row[6].ToString() == textBox1.Text)

{

label6.Text = row[1].ToString();

label7.Text = row[2].ToString();

label24.Text = row[3].ToString();

label26.Text = row[4].ToString();

label28.Text = row[5].ToString().Substring(0, 10);

label9.Text = row[6].ToString();

label19.Text = row[9].ToString();

label11.Text = row[10].ToString();

label13.Text = row[16].ToString();

label15.Text = row[7].ToString() + " руб.";

label17.Text = row[11].ToString();

label22.Text = Convert.ToBoolean(row[13])? "Да" : "Нет";

label31.Text = Convert.ToInt32(row[14])==3? "Да" : "Нет";

label22.ForeColor = Convert.ToBoolean(row[13])? Color.LimeGreen : Color.DarkRed;

label31.ForeColor = Convert.ToInt32(row[14]) == 3 ? Color.LimeGreen : Color.DarkRed;

label4.Visible = false;

button2.Top = 123;

button1.Top = 169;

button3.Top = 215;

return;

}

else

{

label3.Visible = false;

label4.Visible = true;

button2.Top = 136;

button1.Top = 182;

button3.Top = 228;

}

}

}

Код нажатия на кнопку «Регистрация карты»:

private void button1\_Click(object sender, EventArgs e)

{

Form4 f = new Form4();

f.Show();

this.Hide();

}

Код нажатия на кнопку «Далее»:

private void button1\_Click(object sender, EventArgs e)

{

label8.Visible = false;

if (textBox3.Text != "" && textBox5.Text != "" && textBox9.Text != "" && textBox7.Text != "" && (textBox6.Text != "" && textBox8.Text != "" || !credit))

{

label3.Visible = true;

label2.Visible = true;

label5.Visible = true;

textBox1.Visible = true;

textBox2.Visible = true;

textBox4.Visible = true;

button1.Visible = false;

comboBox1.Enabled = false;

this.Height = credit ? 529 : 371;

Random rand = new Random();

textBox4.Text += rand.Next(100, 999).ToString();

for (int i = 0; i < 16; i++) textBox1.Text += rand.Next(0, 9).ToString();

}

else

{

label8.Visible = true;

}

}

Код при выборе типа карты из поля «Тип карты»:

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

if(comboBox1.SelectedIndex != 0)

{

textBox6.Text = "";

label3.Top = 379;

label5.Top = 379;

textBox1.Top = 402;

textBox4.Top = 402;

label2.Top = 444;

textBox2.Top = 440;

label9.Top = 427;

button2.Top = 444;

this.Height = 478;

label8.Top = 379;

button1.Top = 396;

label10.Visible = true;

textBox6.Visible = true;

label12.Visible = true;

numericUpDown1.Visible = true;

label14.Visible = true;

textBox8.Visible = true;

textBox10.Visible = true;

textBox11.Visible = true;

label16.Visible = true;

label17.Visible = true;

label18.Visible = true;

label19.Visible = true;

label20.Visible = true;

label21.Visible = true;

credit = true;

}

else

{

textBox6.Text = "0";

label3.Top = 224;

label5.Top = 224;

textBox1.Top = 247;

textBox4.Top = 247;

label2.Top = 289;

textBox2.Top = 285;

label9.Top = 272;

button2.Top = 289;

this.Height = 324;

label8.Top = 224;

button1.Top = 241;

label10.Visible = false;

textBox6.Visible = false;

label12.Visible = false;

numericUpDown1.Visible = false;

label14.Visible = false;

textBox8.Visible = false;

textBox10.Visible = false;

textBox11.Visible = false;

label16.Visible = false;

label17.Visible = false;

label18.Visible = false;

label19.Visible = false;

label20.Visible = false;

label21.Visible = false;

credit = false;

}

}

Код нажатия на кнопку «Отправить заявку»:

private void button2\_Click(object sender, EventArgs e)

{

button2.Top = 334;

label9.Visible = false;

Form2 f2 = new Form2();

if (textBox2.Text != "")

{

bool there = false;

label4.Visible = false;

string get\_users = "SELECT \* FROM cards ";

SqlDataAdapter da = new SqlDataAdapter(get\_users, Program.f1.ConnStr);

DataSet ds = new DataSet();

da.Fill(ds, "[cards]");

DataTable dt = ds.Tables[0];

foreach (DataRow row in dt.Rows)

{

var login = row[0].ToString();

if (login == textBox1.Text)

{

there = true;

label4.Visible = true;

return;

}

}

if (!there)

{

string addClient = "INSERT INTO clients ([FIO], [number\_passport], [pass\_issued\_by], [tel], [DOB]) VALUES ('"

+ textBox3.Text + "', '"

+ textBox5.Text + "', '"

+ textBox9.Text + "', '"

+ textBox7.Text + "', '"

+ dateTimePicker1.Value + "')";

MyExecuteNonQuery(addClient);

string get\_client = "SELECT \* FROM clients ORDER BY id\_client DESC";

da = new SqlDataAdapter(get\_client, Program.f1.ConnStr);

da.Fill(ds, "[clients]");

string addCard = "INSERT INTO cards ([number\_card], [money], [id\_type], [date], [cvv], [pin], [id\_client], [approved], [made]) VALUES ('"

+ textBox1.Text + "', '"

+ 0 + "', '"

+ (comboBox1.SelectedIndex + 1) + "', '"

+ DateTime.Now.ToString("MM") + "/" + (Convert.ToInt32(DateTime.Now.ToString("yy")) + 4) + "', '"

+ textBox4.Text + "', '"

+ textBox2.Text + "', '"

+ ds.Tables[1].Rows[0][0] + "', '"

+ false + "', '"

+ 1 + "')";

MyExecuteNonQuery(addCard);

string addContract = "INSERT INTO contracts ([id\_client], [date\_of\_conclusion], [amount], [term]) VALUES ('"

+ ds.Tables[1].Rows[0][0] + "', '"

+ DateTime.Now + "', '"

+ numericUpDown1.Value + "', '"

+ textBox8.Text + "')";

MyExecuteNonQuery(addContract);

string addOrder = "INSERT INTO orders ([number\_card], [info\_loan], [amount], [term], [id\_user]) VALUES ('"

+ textBox1.Text + "', '"

+ textBox6.Text + "', '"

+ numericUpDown1.Value + "', '"

+ textBox8.Text + "', '"

+ Program.f1.GetUser() + "')";

MyExecuteNonQuery(addOrder);

string updateProcessed = "UPDATE users SET processed = processed + 1 WHERE (id\_user = '"+Program.f1.GetUser()+"')";

MyExecuteNonQuery(updateProcessed);

f2.Show();

this.Hide();

}

}

else

{

label9.Visible = true;

}

}

Нажатие на кнопку «Клиенты»:

private void button3\_Click(object sender, EventArgs e)

{

Form5 f5 = new Form5();

f5.Show();

this.Hide();

}

Кнопка «Фильтр»:

private void button3\_Click(object sender, EventArgs e)

{

string CommandText = "SELECT " +

"clients.FIO as [ФИО], " +

"cards.number\_card as [Номер карты], " +

"clients.number\_passport as [Номер паспорта], " +

"cards.date as [Действие карты], " +

"types.type\_name as [Тип карты], " +

"cards.made, " +

"mades.name as [Изготовление] " +

"FROM clients, cards, types, mades " +

"WHERE (clients.id\_client = cards.id\_client) " +

"AND (types.id\_type = cards.id\_type) " +

"AND (cards.made = mades.made)";

if (textBox1.Text != "")

{

if (comboBox1.SelectedIndex == 0)

CommandText = CommandText + " AND ([clients].[FIO] LIKE '%" + textBox1.Text + "%')";

if (comboBox1.SelectedIndex == 1)

CommandText = CommandText + " AND ([cards].[number\_card] LIKE '%" + textBox1.Text + "%') ";

if (comboBox1.SelectedIndex == 2)

CommandText = CommandText + " AND ([clients].[number\_passport] LIKE '%" + textBox1.Text + "%') ";

if (comboBox1.SelectedIndex == 3)

CommandText = CommandText + " AND ([cards].[date] LIKE '%" + textBox1.Text + "%') ";

}

SqlDataAdapter da = new SqlDataAdapter(CommandText, Program.f1.ConnStr);

DataSet ds = new DataSet();

da.Fill(ds, "[clients]");

dataGridView1.DataSource = ds.Tables["[clients]"].DefaultView;

}

Код нажатия на кнопку «Отследить»:

private void button4\_Click(object sender, EventArgs e)

{

int index = dataGridView1.CurrentRow.Index;

Form9 f9 = new Form9(Convert.ToString(dataGridView1[1, index].Value));

f9.Show();

}

Код при загрузки формы «Отслеживание изготовления»:

private void Form9\_Load(object sender, EventArgs e)

{

timer1.Start();

progressHundler();

}

private void progressHundler(){

string get\_card = "SELECT production\_order.application\_date, clients.FIO FROM cards, production\_order, clients " +

"WHERE (production\_order.number\_card = '" + num\_card + "') " +

"AND (production\_order.number\_card = cards.number\_card) " +

"AND (cards.id\_client = clients.id\_client)";

SqlDataAdapter da = new SqlDataAdapter(get\_card, Program.f1.ConnStr);

DataSet ds = new DataSet();

da.Fill(ds, "[cards]");

label2.Text = ds.Tables[0].Rows[0][1].ToString();

label3.Text = num\_card;

DateTime begin = Convert.ToDateTime(ds.Tables[0].Rows[0][0]);

double full = (begin - Convert.ToDateTime(ds.Tables[0].Rows[0][0]).AddMinutes(10)).TotalSeconds;

double current = (begin - DateTime.Now).TotalSeconds;

int result = Convert.ToInt32((current \* 100) / full);

if (result< 101) progressBar1.Value = result;

if (result >= 1 && result< 10) label6.Text = "Сейчас, будущая пластиковая карта, проходит процесс печати через трафарет.";

if (result >= 10 && result< 20) label6.Text = "Пластиковая карта проходит этап сборки и ламинирования.";

if (result >= 20 && result< 25) label6.Text = "Пластиковая карта проходит этап вырубки. Теперь она приобретает типичный, для пластиковых карт, вид.";

if (result >= 25 && result< 35) label6.Text = "Пластиковая карта переносится в цех тиснения и имплантации. На карту наносится полоса подписи, защитная голограмма и фрезеруется место под чип";

if (result >= 35 && result< 45) label6.Text = "Пластиковая карта проходит лабораторные испытания. Карту испытывают и проверяют на прочность.";

if (result >= 45 && result< 55) label6.Text = "Пластиковая карта переносится в цех эмбоссирования. Банк передаёт данные для персонализации клиентов, карты чипируют и осуществляют эмбоссирование.";

if (result >= 55 && result< 60) label6.Text = "Упаковка карт. Подготовка к отправке в банк.";

if (result >= 60 && result< 75) label6.Text = "Карту доставляют из цеха в центральный банк.";

if (result >= 75 && result< 80) label6.Text = "Карта доставлена в центральный банк. Подготовка к отправке в отделение.";

if (result >= 80 && result< 95) label6.Text = "Карту доставляют из центрального банка в отделение.";

if (result >= 95 && result< 100) label6.Text = "Карта доставлена в отделение. Проходит процесс распаковки.";

if (result > 100) {

label6.Text = "Перезапустите программу.";

timer1.Start();

}

}

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

{

progressHundler();

}

Код кнопки «Удалить» на форме с клиентами:

private void button2\_Click(object sender, EventArgs e)

{

int index = dataGridView1.CurrentRow.Index;

string get\_card = "SELECT cards.id\_client FROM cards WHERE ([cards].[number\_card] = '" + Convert.ToString(dataGridView1[1, index].Value) + "')";

SqlDataAdapter da = new SqlDataAdapter(get\_card, Program.f1.ConnStr);

DataSet ds = new DataSet();

da.Fill(ds, "[cards]");

string DeleteOrderProd = "DELETE FROM production\_order WHERE ([production\_order].[number\_card] = '" + Convert.ToString(dataGridView1[1, index].Value) + "')";

string DeleteCard = "DELETE FROM cards WHERE ([cards].[number\_card] = '" + Convert.ToString(dataGridView1[1, index].Value) + "')";

string DeleteContract = "DELETE FROM contracts WHERE ([contracts].[id\_client] = '" + ds.Tables[0].Rows[0][0] + "')";

string DeleteClient = "DELETE FROM clients WHERE ([clients].[id\_client] = '" + ds.Tables[0].Rows[0][0] + "')";

dataGridView1.Rows.RemoveAt(index);

MyExecuteNonQuery(DeleteOrderProd);

MyExecuteNonQuery(DeleteCard);

MyExecuteNonQuery(DeleteContract);

MyExecuteNonQuery(DeleteClient);

}

Код кнопки «Подробно» на форме с клиентами:

private void button1\_Click(object sender, EventArgs e)

{

Form2 f = new Form2();

int index = dataGridView1.CurrentRow.Index;

string CommandText = "SELECT " +

"contracts.id\_contract " +

"FROM clients, cards, contracts " +

"WHERE (cards.number\_card = '"+Convert.ToString(dataGridView1[1, index].Value)+"') " +

"AND (cards.id\_client = clients.id\_client) " +

"AND (contracts.id\_client = clients.id\_client)";

SqlDataAdapter da = new SqlDataAdapter(CommandText, Program.f1.ConnStr);

DataSet ds = new DataSet();

da.Fill(ds, "[clients]");

f.Show();

f.getInfoSelect(Convert.ToString(dataGridView1[1, index].Value), ds.Tables[0].Rows[0][0].ToString());

this.Hide();

}

Код кнопки «Сформировать клиентский договор» на основной форме консультанта:

private void button4\_Click(object sender, EventArgs e)

{

string getClient = "SELECT \* FROM Contracts " +

"WHERE ([Contracts].[id\_contract] = '" + id\_contract + "')";

SqlDataAdapter da = new SqlDataAdapter(getClient, Program.f1.ConnStr);

DataSet ds = new DataSet();

da.Fill(ds, "[Contract]");

string pathDocument = Environment.GetFolderPath(Environment.SpecialFolder.Desktop) + "\\Кредитный\_документ№" + id\_contract + ".docx";

//string pathDocument = "\\путь\\ещёпуть\\папкакакаянибудь\\иещёпапка210препод\\Документ№" + dataGridView1[0, index].Value.ToString() + ".docx";

DocX document = DocX.Create(pathDocument);

document.InsertParagraph("Кредитный договор №" + id\_contract).

Font("Times New Roman").

FontSize(16).

Color(Color.Black).

Bold().

Alignment = Alignment.center;

Paragraph paragraph = document.InsertParagraph();

paragraph.AppendLine(Convert.ToDateTime(ds.Tables[0].Rows[0][2].ToString().Substring(0, 10)).ToString("dd MMMM yyyy") + "г.").

FontSize(12).

Font("Times New Roman").

Italic().

UnderlineStyle(UnderlineStyle.singleLine).

Alignment = Alignment.right;

paragraph.AppendLine();

Paragraph intro = document.InsertParagraph();

intro.Append("ООО «Главбанк»‎ ").FontSize(14).Font("Times New Roman").Italic().UnderlineStyle(UnderlineStyle.singleLine).Alignment = Alignment.both;

intro.Append("в лице ").FontSize(14).Font("Times New Roman");

intro.Append("Борова Степана Дмитриевича ").FontSize(14).Font("Times New Roman").UnderlineStyle(UnderlineStyle.singleLine).Italic();

intro.Append(" действующего на основании Устава, именуемый в дальнейшем ").FontSize(14).Font("Times New Roman");

intro.Append("«Кредитор»").Bold().FontSize(14).Font("Times New Roman");

intro.Append(", с одной стороны, и ").FontSize(14).Font("Times New Roman");

intro.Append(label6.Text).FontSize(14).Font("Times New Roman").UnderlineStyle(UnderlineStyle.singleLine).Italic();

intro.Append(", действующего на основании Устава, именуемый в дальнейшем ").FontSize(14).Font("Times New Roman");

intro.Append("«Заёмщик»‎").FontSize(14).Font("Times New Roman");

intro.Append(", с другой стороны, именуемые в дальнейшем ").FontSize(14).Font("Times New Roman");

intro.Append("«Стороны»").Bold().FontSize(14).Font("Times New Roman");

intro.Append(", заключили настоящий договор, в дальнейшем ").FontSize(14).Font("Times New Roman");

intro.Append("«Договор»").Bold().FontSize(14).Font("Times New Roman");

intro.Append(", о нижеследующем:").FontSize(14).Font("Times New Roman");

Paragraph main\_part = document.InsertParagraph();

main\_part.AppendLine("Статья 1. Предмет Договора").

Font("Times New Roman").

FontSize(14).

Bold().

Alignment = Alignment.left;

main\_part.AppendLine("1.1. Общая размера кредита по настоящему Договору составит " + ds.Tables[0].Rows[0][3].ToString() + " руб. ").FontSize(14).Font("Times New Roman");

main\_part.AppendLine("1.2. Заёмщик берет кредит на условиях, предусмотренных настоящим Договором и обязуется соблюдать" +

" условия предоставления кредита, а также возвратить полученную денежную сумму и уплатиьт проценты на неё в размере, сроки и на условиях" +

", предусмотренных настоящим Договором.").FontSize(14).Font("Times New Roman");

main\_part.AppendLine("1.3. Кредит предоствляется на срок с " + Convert.ToDateTime(ds.Tables[0].Rows[0][2].ToString().Substring(0, 10)).ToString("dd MMMM yyyy") + " " +

"года (\"Дата предоставления кредита\") по " + Convert.ToDateTime(ds.Tables[0].Rows[0][2].ToString().Substring(0, 10)).AddMonths(Convert.ToInt32(ds.Tables[0].Rows[0][4])).ToString("dd MMMM yyyy") + " " +

"года включительно (\"Дата погошения кредита\")").FontSize(14).Font("Times New Roman");

main\_part.AppendLine("1.4. Условием возникновения прав и обязанностей Сторон по настоящему Договору является вступление в силу договора залога.").FontSize(14).Font("Times New Roman");

main\_part.AppendLine();

main\_part.AppendLine("Статья 2. Проценты").

Font("Times New Roman").

FontSize(14).

Bold().

Alignment = Alignment.left;

main\_part.AppendLine("2.1. За пользование кредитом Заемщик уплачивает Кредитору 13% (тринадцать) процентов годовых на сумму кредита" +

", указанную в п.1.1 настоящего Договора").FontSize(14).Font("Times New Roman");

main\_part.AppendLine("2.2. Размер процентной ставки за пользование кредитом может быть изменен Кредитором в одностороннем порядке в связи с " +

"изменением ставки рефинансирования ЦБ РФ").FontSize(14).Font("Times New Roman");

main\_part.AppendLine("2.3. При изменении процентной ставки за пользование кредитом Кредитор извещает об этом Заёмщика заказным письмом " +

"в 3-дневный срок с момента принятия решения об изменении размера процентной ставки по адресу," +

" указанному в настоящем Договоре").FontSize(14).Font("Times New Roman");

main\_part.AppendLine();

Paragraph outro = document.InsertParagraph();

outro.AppendLine("Статья 3. Юридические реквизиты сторон").

Font("Times New Roman").

FontSize(14).

Color(Color.Black).

Bold().

Alignment = Alignment.left;

Table table = document.AddTable(8, 3);

table.Rows[0].Cells[1].Paragraphs[0].Append("Кредитор").Font("Times New Roman").FontSize(12).Bold();

table.Rows[0].Cells[2].Paragraphs[0].Append("Заемщик").Font("Times New Roman").FontSize(12).Bold();

table.Rows[1].Cells[0].Paragraphs[0].Append("ФИО").Font("Times New Roman").FontSize(12).Bold().Alignment = Alignment.right;

table.Rows[1].Cells[1].Paragraphs[0].Append("Боров Степан Дмитриевич").Font("Times New Roman").FontSize(12);

table.Rows[1].Cells[2].Paragraphs[0].Append(label6.Text).Font("Times New Roman").FontSize(12);

table.Rows[2].Cells[0].Paragraphs[0].Append("Телефон").Font("Times New Roman").FontSize(12).Bold().Alignment = Alignment.right;

table.Rows[2].Cells[1].Paragraphs[0].Append("89237164835").Font("Times New Roman").FontSize(12);

table.Rows[2].Cells[2].Paragraphs[0].Append(label26.Text).Font("Times New Roman").FontSize(12);

table.Rows[3].Cells[0].Paragraphs[0].Append("Номер карты").Font("Times New Roman").FontSize(12).Bold().Alignment = Alignment.right;

table.Rows[3].Cells[1].Paragraphs[0].Append("7394000116273895").Font("Times New Roman").FontSize(12);

table.Rows[3].Cells[2].Paragraphs[0].Append(label9.Text).Font("Times New Roman").FontSize(12);

table.Rows[4].Cells[0].Paragraphs[0].Append("Номер паспорта").Font("Times New Roman").FontSize(12).Bold().Alignment = Alignment.right;

table.Rows[4].Cells[1].Paragraphs[0].Append("1625849475").Font("Times New Roman").FontSize(12);

table.Rows[4].Cells[2].Paragraphs[0].Append(label7.Text).Font("Times New Roman").FontSize(12);

table.Rows[6].Cells[1].Paragraphs[0].Append("Подпись: ").Font("Times New Roman").FontSize(12);

table.Rows[6].Cells[2].Paragraphs[0].Append("Подпись: ").Font("Times New Roman").FontSize(12);

document.InsertParagraph().InsertTableAfterSelf(table);

document.Save();

label30.Text = "Документ сформирован и сохранён на рабочий стол.";

}

Переходим в окно авторизации и входим под учетную запись кредитного специалиста. Код кнопки «Начать работу»:

private void button2\_Click(object sender, EventArgs e)

{

Form8 f8 = new Form8();

id\_order = dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells[0].Value.ToString();

f8.Show();

this.Hide();

}

Код загрузки формы кредитного специалиста:

private void Form8\_Load(object sender, EventArgs e)

{

string SqlText = "SELECT " +

"clients.FIO, " +

"clients.DOB, " +

"clients.number\_passport, " +

"clients.pass\_issued\_by, " +

"cards.number\_card, " +

"types.type\_name, " +

"clients.tel, " +

"orders.amount, " +

"orders.term, " +

"orders.info\_loan, " +

"clients.id\_client " +

"FROM orders, cards, types, clients " +

"WHERE (orders.id\_order = '"+ Program.f7.GetIdOrder() + "')" +

"AND (cards.number\_card = orders.number\_card)" +

"AND (cards.id\_client = clients.id\_client)" +

"AND (cards.id\_type = types.id\_type)";

SqlDataAdapter da = new SqlDataAdapter(SqlText, Program.f1.ConnStr);

da.Fill(ds, "[clients]");

label9.Text = ds.Tables[0].Rows[0][0].ToString();

label10.Text = ds.Tables[0].Rows[0][1].ToString();

label12.Text = ds.Tables[0].Rows[0][2].ToString();

label14.Text = ds.Tables[0].Rows[0][3].ToString();

label16.Text = ds.Tables[0].Rows[0][4].ToString();

label18.Text = ds.Tables[0].Rows[0][5].ToString();

label21.Text = ds.Tables[0].Rows[0][6].ToString();

if(ds.Tables[0].Rows[0][7].ToString() != "0" && ds.Tables[0].Rows[0][8].ToString() != "0" && ds.Tables[0].Rows[0][9].ToString() != "0")

{

label7.Text = ds.Tables[0].Rows[0][7].ToString()+" рублей";

label8.Text = ds.Tables[0].Rows[0][8].ToString()+" месяцев";

label20.Text = ds.Tables[0].Rows[0][9].ToString();

button2.Visible = true;

label1.Visible = true;

label2.Visible = true;

label23.Visible = false;

}

else

{

label7.Text = "Отсутствует";

label8.Text = "Отсутствует";

label20.Text = "Отсутствует";

button2.Visible = false;

label1.Visible = false;

label2.Visible = false;

label23.Visible = true;

}

}

Код кнопки «Одобрить»:

private void button1\_Click(object sender, EventArgs e)

{

string updateProcessed = "UPDATE users SET processed = processed + 1 WHERE (id\_user = '" + Program.f1.GetUser() + "')";

MyExecuteNonQuery(updateProcessed);

string addOrder = "INSERT INTO production\_order ([number\_card], [application\_date]) VALUES ('"

+ ds.Tables[0].Rows[0][4].ToString() + "', '"

+ DateTime.Now + "')";

MyExecuteNonQuery(addOrder);

string updateApproved = "UPDATE cards SET approved = 'True', money = '" + ds.Tables[0].Rows[0][7].ToString() + "', made = 2 WHERE (number\_card = '" + ds.Tables[0].Rows[0][4].ToString() + "')";

MyExecuteNonQuery(updateApproved);

string DeleteOrder = "DELETE FROM orders WHERE (orders.number\_card = '" + ds.Tables[0].Rows[0][4].ToString() + "')";

MyExecuteNonQuery(DeleteOrder);

Form7 f7 = new Form7();

f7.Show();

this.Hide();

}

Код кнопки «Отклонить»:

private void button3\_Click(object sender, EventArgs e)

{

string DeleteOrder = "DELETE FROM orders WHERE (orders.number\_card = '" + ds.Tables[0].Rows[0][4].ToString() + "')";

MyExecuteNonQuery(DeleteOrder);

string DeleteCard = "DELETE FROM cards WHERE ([cards].[number\_card] = '" + ds.Tables[0].Rows[0][4].ToString() + "')";

string DeleteContract = "DELETE FROM contracts WHERE ([contracts].[id\_client] = '" + ds.Tables[0].Rows[0][10].ToString() + "')";

string DeleteClient = "DELETE FROM clients WHERE ([clients].[id\_client] = '" + ds.Tables[0].Rows[0][10].ToString() + "')";

MyExecuteNonQuery(DeleteCard);

MyExecuteNonQuery(DeleteContract);

MyExecuteNonQuery(DeleteClient);

Form7 f7 = new Form7();

f7.Show();

this.Hide();

}

Код кнопки «Отправить запрос в БКИ»:

private void button2\_Click(object sender, EventArgs e)

{

label2.ForeColor = SystemColors.Control;

label2.Text = "На рассмотрении...";

timer1.Start();

}

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

{

Random r = new Random();

int act = r.Next(0, 100);

if (act < 90)

{

label2.ForeColor = Color.LimeGreen;

label2.Text = "Ответ положительный";

}

else

{

label2.ForeColor = Color.Tomato;

label2.Text = "Ответ отрицательный";

}

timer1.Stop();

}

# ГЛАВА 4. ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЙ СИСТЕМ

Тестирование информационной системы – это один из ключевых этапов разработки, требующий самого пристального внимания.

Таблица 8 – Общая информация о тестировании

|  |  |
| --- | --- |
| Название проекта | Оформление банковских карт |
| Номер версии | 01 |
| Имя тестера | Главбанк |
| Даты тестирования | 20.06.2021 |

Таблица 9 – Test case #1

|  |  |
| --- | --- |
| Test Case # | Главбанк |
| Приоритет теста | *высокий* |
| 1)Название тестирования/Имя | Проверка корректности ввода логина и пароля на странице авторизации |
| Резюме испытания | Открытие страницы администратора, консультанта, кредитного специалиста |
| Шаги тестирования | 1) в поле «Логин» ввести имя пользователя;  2) в поле «Пароль» ввести пароль пользователя;  3) нажать на кнопку «Войти» |
| Данные тестирования | Набор #1 (правильный - администратор):  1) «Логин» = 2;  2) «Пароль» = 2.  Набор #2 (правильный - консультант):  1) «Логин» = «519»;  2) «Пароль» = «519».  Набор #3 (правильный – кредитный специалист):  1) «Логин» = «qwer»;  2) «Пароль» = «qwer».  Набор #4 (неправильный - консультант):  1) «Логин» = «516»;  2) «Пароль» = «516». |

Продолжение таблицы 9

|  |  |
| --- | --- |
| Ожидаемый результат | Набор #1: открывается панель администратора;  Набор #2: открывается панель консультанта;  Набор #3: открывается панель кредитного специалиста;  Набор #4: на форме выводится сообщение «Неверный пароль или логин», страница авторизации остается активной |
| Фактический результат | Набор #1: открывается панель администратора;  Набор #2: открывается панель консультанта;  Набор #3: открывается панель кредитного специалиста;  Набор #4: на форме выводится сообщение «Неверный пароль или логин», страница авторизации остается активной |
| Предпосылки | Ввод данных не зависит от языка |
| Постусловия | Набор #1: активным является страница администратора;  Набор #2: активной является страница консультанта;  Набор #3: активной является страница кредитного специалиста;  Набор #4: активной является страница кредитного специалиста. |
| Статус (Pass/Fail) | Pass |
| Комментарии | При вводе логина и пароля регистр нужно учитывать |

Таблица 10 – Test case #2

|  |  |
| --- | --- |
| 2) Название тестирования/Имя | Регистрация пользователя |
| Резюме испытания | Регистрация пользователя в систему |

Продолжение таблицы 10

|  |  |
| --- | --- |
| Шаги тестирования | После авторизации под администратором, нажимаем кнопку «Добавить сотрудника»  1) В поле «ФИО» ввести ФИО пользователя;  2) В списке «Роль» выбрать нужную роль;  3) В поле «Логин» ввести логин;  4) В поле «Пароль» ввести пароль;  5) Нажать на кнопку «Зарегистрировать» |
| Данные тестирования | Набор #1 (правильный):  1) «ФИО» = «Сирина Елена Ивановна»;  2) «Роль» = «Кредитный специалист»;  3) «Логин» = «elena123»;  4) «Пароль» = «jFJf67FF».  Набор #2 (неправильный):  1) «ФИО» = «Сирина Елена Ивановна»;  2) «Роль» = «Кредитный специалист»;  3) «Логин» = «qwer»;  4) «Пароль» = «qwer». |
| Ожидаемый результат | Набор #1: Пользователь зарегистрирован в систему;  Набор #2: В форме выводится сообщение «Такой логин уже существует», страница регистрации остается активной |
| Фактический результат | Набор #1: Пользователь зарегистрирован в систему;  Набор #2: В форме выводится сообщение «Такой логин уже существует», страница регистрации остается активной |
| Предпосылки | Ввод данных не зависит от языка |
| Постусловия | Набор #1: Активной является страница администратора;  Набор #2: Активной является страница регистрации. |

Продолжение таблицы 10

|  |  |
| --- | --- |
| Статус (Pass/Fail) | pass |

Таблица 11 – Test case #3

|  |  |
| --- | --- |
| 3) Название тестирования/Имя | Удаление данных |
| Резюме испытания | Удаление данных из таблиц |
| Шаги тестирования | 1) проходим авторизацию под администратором;  2) выбираем нужную строку;  3) нажимаем кнопку «Удалить». |
| Ожидаемый результат | Данные в таблице обновляются. |
| Фактический результат | Данные в таблице обновляются. |
| Постусловия | Активным является панель администратора. |
| Статус (Pass/Fail) | Pass |

Проведенное тестирование программы позволяет сделать вывод о работоспособности программы и ее соответствии заданию.

Выявленные недостатки предполагают дальнейшую работу над совершенствованием программы, что предоставляет обширное поле деятельности для повышения знаний и совершенствовании навыков программирования на С#, освоении новых аспектов программирования в современной среде .NET.

Тестирование показало корректную работу разработанной базы данных.

# ГЛАВА 5. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Руководство предназначено для ознакомления пользователя с техническими характеристиками и функциональными возможностями автоматизированной информационной системы.

При запуске программы откроется окно с главной формой программы — «Авторизация» (в соответствии с Рисунок 6-форма авторизации сотрудников). На данной форме всем сотрудникам нужно ввести логин и пароль, далее нажать кнопку «Войти».

Администратор – пользователь, который управляет сотрудниками и следит за их работой. Для администратора доступна соответствующая форма (рис. 7), которая обеспечивает все условия для осуществления работы. Администратор может удалять сотрудников и добавлять их (рис. 8). Также для администратора доступна вся информация о всех сотрудниках отделения.

Консультант – пользователь, который занимается оформлением банковских карт и просматривает всю необходимую информацию о клиентах и их составляющих. Для консультанта доступна главная панель (рис 10), где можно зарегистрировать новую карту, через соответствующую форму (рис. 12) нажав на кнопку «Регистрация карты». Также через главную панель можно перейти на форму с таблицей клиентов отделения (рис. 15), нажав на кнопку «Клиенты». На этой форме можно: удалять клиентов, нажав на кнопку «Удалить», просматривать подробную информацию, нажав на кнопку «Подробно», отследить процесс изготовления карты, нажав на кнопку «Отследить». Отслеживание изготовления проходит в отдельной форме (рис. 16).

Кредитный специалист – пользователь, который занимается одобрением заявок на банковскую карту. Главная панель кредитного отдела является форма с таблицей всех заявок от консультанта (рис. 17). Кредитному специалисту нужно выбрать любую заявку и нажать кнопку «Начать работу», чтобы открылась форма (рис. 18), в которой сотрудник может просмотреть информацию о данном специалисте и сделать соответствующий вывод в разделе «Итог».

# ЗАКЛЮЧЕНИЕ

В результате была модифицирована информационная система для оформления банковских карт. Данная система имеет удобный пользовательский интерфейс, позволяющий легко освоить работу в программе, гибкость же программного кода в случае необходимости позволит удовлетворить растущие требования к системе.

Преследуемая изначально цель автоматизации ведения базы данных, для оформления банковских карт, достигнута.

Разработанную автоматизированную систему можно интегрировать в отделения банка.

Разработанная система позволит повысить скорость обработки информации и сэкономит время работы пользователей.

Заказчик программного продукта получает полный доступ к исходным данным.

В информационной системе оформления банковских карт, в режиме Администратора возможно выполнение следующих действий:

* добавление сотрудников;
* удаления сотрудников;
* просмотр и поиск записей.

В режиме Консультанта возможно выполнение следующих действий:

* оформление заявки;
* просмотр подробной информации о клиенте;
* просмотр и поиск записей о клиентах;
* удаление клиентов;
* отслеживание процесса изготовления карты (имитация);
* формирование договора в формате DOCX.

В режиме Кредитного Специалиста возможно выполнение следующих действий:

* просмотр подробной информации о клиенте;
* отправка запроса в отдел БКИ (имитация);
* формирование заявки на изготовление.

Программа требует минимальные ресурсы компьютера. Программный продукт можно переносить на различных источниках. После внедрения возможны изменения, доработки. Это все зависит от пожеланий заказчика.

# СПИСОК ИСПОЛЬЗУВАЕМЫХ ИСТОЧНИКОВ

1. ГОСТ РД 50-34.698-90 Руководство пользователя
2. ГОСТ 2.001-2013 ЕСПД Единая система конструкторской документации
3. ГОСТ 12.0.004-2015 "ССБТ Охрана труда
4. ГОСТ Р 43.0.11-2014 Базы данных в технической деятельности
5. Вишневский, Алексей Microsoft SQL Server. Эффективная работа / Алексей Вишневский. - М.: Питер, 2017. - 143 c.
6. Голицына, О.Л. Базы данных: Учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. - М.: Форум, 2017. - 400 c.
7. Жилинский. А. Самоучитель Microsoft SQL Server/ А. Жилинский. - М.: БХВ-Петербург, 2016. - 224 c.
8. Кузин, А.В. Базы данных: Учебное пособие для студ. высш. учеб. заведений / А.В. Кузин, С.В. Левонисова. - М.: ИЦ Академия, 2012. - 320 c.
9. Мартишин С. А. Базы данных. Практическое применение СУБД SQL и NoSOL-типа для применения проектирования информационных систем: Учебное пособие. - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2017. - 368 с.
10. Редько, В.Н. Базы данных и информационные системы / В.Н. Редько, И.А. Бассараб. - М.: Знание, 2018. - 240 c.
11. Ульман, Дж. Основы систем баз данных / Дж. Ульман. - М.: Финансы и статистика, 2017. - 292 c.
12. Федорова Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: учебное пособие. — М.: КУРС: ИНФРА-М, 2017. — 336 с.
13. Шустова Л.И. Базы данных: учебник / Л.И. Шустова, О.В. Тараканов. — М.: ИНФРА-М, 2017. — 304 с.
14. ГАЗПРОМБАНК <https://gazprombank.ru/> (17.06.2021)
15. РНКБ <https://rncb.ru/> (17.06.2021)

Краевое государственное автономное

профессиональное образовательное учреждение

«Ачинский техникум нефти и газа»

**ОТЗЫВ**

**на выполнение курсовой работы**

|  |  |  |
| --- | --- | --- |
| ФИО студента |  | Назаров Андрей Андреевич |
|  |  |  |
| Тема курсовой работы |  | Модификация информационной системы «Оформление банковских карт» |
|  |
|  | | | |

1. Системно и последовательно ли работал студент над заявленной темой?

|  |  |
| --- | --- |
|  | Да |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  | Нет |

1. Проявлял ли самостоятельность, творчество в процессе работы?

|  |  |
| --- | --- |
|  | Да |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  | Нет |

1. Какова глубина проработки темы курсовой работы (проекта)?

|  |  |
| --- | --- |
|  | Оптимально |
|  | Достаточно |
|  | Недостаточно |

1. Уровень выполнения практической части работы?

|  |  |
| --- | --- |
|  | Оптимально |
|  | Достаточно |
|  | Недостаточно |

1. Представил ли материалы, подтверждающие его практическую деятельность?

|  |  |
| --- | --- |
|  | Да |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  | Нет |

1. Своевременно ли выполнял работу согласно этапам графику?

|  |  |
| --- | --- |
|  | Да |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  | Нет |

**Показал ли студент при выполнении курсовой работы (проекта) сформированность следующих общих и профессиональных компетенций:**

1. Понимание сущности и социальной значимости своей будущей профессии, устойчивый интерес к ней:

|  |  |
| --- | --- |
|  | Оптимально |
|  | Достаточно |
|  | Недостаточно |

1. Умение организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

|  |  |
| --- | --- |
|  | Оптимально |
|  | Достаточно |
|  | Недостаточно |

1. Умение осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития:

|  |  |
| --- | --- |
|  | Оптимально |
|  | Достаточно |
|  | Недостаточно |

1. Владение информационной культурой, умение анализировать и оценивать информацию с использованием информационно-коммуникационных технологий:

|  |  |
| --- | --- |
|  | Оптимально |
|  | Достаточно |
|  | Недостаточно |

1. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием:

|  |  |
| --- | --- |
|  | Оптимально |
|  | Достаточно |
|  | Недостаточно |

1. Осуществлять профессиональную деятельность в условиях обновления ее целей, содержания, смены технологий, изменения нормативно-правовой базы:

|  |  |
| --- | --- |
|  | Оптимально |
|  | Достаточно |
|  | Недостаточно |

1. Решать проблемы, оценивать риски и принимать решения в нестандартных ситуациях:

|  |  |
| --- | --- |
|  | Оптимально |
|  | Достаточно |
|  | Недостаточно |

1. Работать в коллективе и команде, эффективно общаться с коллегами, руководством, потребителями:

|  |  |
| --- | --- |
|  | Оптимально |
|  | Достаточно |
|  | Недостаточно |

1. Брать на себя ответственность за работу членов команды (подчиненных), результат выполнения заданий:

|  |  |
| --- | --- |
|  | Оптимально |
|  | Достаточно |
|  | Недостаточно |

1. Соответствует ли работа требованиям, предъявляемым к курсовой работе по специальности:

|  |  |
| --- | --- |
|  | Да |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  | Нет |

1. Заключение:

соответствие курсовой работы (проекта) заявленной теме\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

оценка полноты проработанного и изложенного материала\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

значимость выполненной курсовой работы (проекта)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

оценка курсовой работы (проекта)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

|  |  |  |  |
| --- | --- | --- | --- |
| Руководитель |  |  | Степанова Ольга Викторовна |
|  | (подпись) |  | (Имя, Отчество Фамилия) |
|  | | | |

Дата