

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика
(код и наименование направления подготовки, специальности)

Бизнес-информатика
(направленность (профиль) / специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему «Автоматизация учета операций кредитования для микрофинансовой
организации»

Обучающийся

И.А. Пахотин

(Инициалы Фамилия)

(личная подпись)

Руководитель

В.Ф. Глазова

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2022

Аннотация

Тема выпускной квалификационной работы: Автоматизация учета операций кредитования для микрофинансовой организации.

Ключевые слова: микрофинансовая организация, автоматизация, кредитные операции, информационная система, язык программирования C#, база данных.

Результатом выпускной квалификационной работы является разработка информационной системы по учету кредитных операций в микрофинансовой организации. Основные функции разработанной системы:

- авторизация под учетными записями различного типа;
- формирование заявок на получение кредита;
- добавление анкетных данных клиентов, а также редактирование и удаление данных по клиентам;
- редактирование заявок и проверка заявок на наличие рисков;
- построение графика выплат;
- формирование документов для заявок с задолженностями;
- построение отчетов по клиентам и заявкам;
- поиск заявок по различным параметрам;
- поиск клиентов по имени, контактными данным.

В первой главе проведен анализ предметной области, построена модель исследуемого процесса, проведен обзор готовых решений, поставлена задача на разработку информационной системы.

Во второй главе проведено проектирование базы данных информационной системы и интерфейса для работы с базой данных.

В третьей главе представлены результаты разработки информационной системы для микрофинансовой организации.

Выпускная квалификационная работа содержит пояснительную записку объемом 69 страниц, включая 34 иллюстрации, 20 таблиц, список использованных источников из 23 наименований, 3 приложения.

Оглавление

Введение.....	5
Глава 1 Функциональное моделирование предметной области.....	7
1.1 Техничко-экономическая характеристика учета операций кредитования в микрофинансовых организациях	7
1.2 Концептуальное моделирование процесса по учету кредитных операций.....	9
1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям	15
1.4 Постановка задачи на разработку системы.....	17
1.5 Разработка модели бизнес-процесса «Как должно быть»	17
Глава 2 Логическое проектирование системы по учету кредитных операций	20
2.1 Выбор технологии логического моделирования информационной системы	20
2.2 Логическая модель системы по учету кредитных операций и ее описание.....	21
2.3 Информационное обеспечение системы	27
2.4 Проектирование базы данных информационной системы по учету кредитных операций	30
2.5 Требования к аппаратно-программному обеспечению системы по учету кредитных операций	34
Глава 3 Физическое проектирование информационной системы по учету кредитных операций	35
3.1 Выбор архитектуры информационной системы по учету кредитных операций	35

3.2	Выбор технологии разработки программного обеспечения информационной системы по учету кредитных операций.....	37
3.3	Выбор системы управления базой данных для информационной системы по учету кредитных операций.....	39
3.4	Разработка физической модели данных информационной системы по учету кредитных операций.....	41
3.5	Разработка программного обеспечения информационной системы по учету кредитных операций	45
3.6	Описание функциональности информационной системы по учету кредитных операций	47
3.7	Оценка и обоснование экономической эффективности разработки информационной системы	60
	Заключение	63
	Список используемой литературы и используемых источников.....	64
	Приложение А Сценарий диалога для информационной системы.....	66
	Приложение Б Дерево программных модулей.....	67
	Приложение В Примеры отчетов	68

Введение

Увеличение потребительской активности граждан и спроса на кредиты и займы на фоне роста инфляции, а также рост доходов способствует активному развитию сектора микрофинансирования.

Развитие указанного сектора в значительной мере обеспечивается увеличением количества займов, а также их средних размеров. Займы выдаются как физическим, так и юридическим лицам. Однако основной вклад в рост сектора микрофинансирования создают именно займы физическим лицам.

Для оценки надежности заемщика сотрудникам микрофинансовых организаций необходимо быстро и точно обрабатывать большие объемы информации. Для того чтобы сделать процесс обработки наиболее оптимальным и сократить количество возможных ошибок, в том числе и по причине человеческого фактора, целесообразно использовать современные информационные технологии.

Использование информационных систем для сбора и обработки информации позволит собрать все данные в едином централизованном хранилище, провести обработку данных любого типа, а также представить результаты обработки в удобном для восприятия виде. Следовательно, разработка информационных систем для учета деятельности микрофинансовых организаций, в частности, для учета кредитных операций, является актуальной задачей.

Объект исследования: процесс учета операций кредитования для микрофинансовой организации.

Предмет исследования: автоматизация процесса учета кредитных операций для микрофинансовой организации.

Целью выпускной квалификационной работы является разработка приложения для автоматизации учета кредитных операций в микрофинансовой организации.

Для достижения указанной цели необходимо решить следующие задачи:

- провести анализ предметной области;
- построить модель процесса по учету кредитных операций в микрофинансовой организации;
- провести обзор готовых решений;
- обосновать необходимость собственной разработки;
- сформулировать задачу на разработку информационной системы;
- спроектировать базу данных для системы учета кредитных операций;
- спроектировать интерфейс к базе данных;
- разработать информационную систему по учету кредитных операций в микрофинансовой организации;
- оценить экономическую эффективность от внедрения системы.

Для решения поставленных задач использовались следующие методы исследования: литературы, материалов и документов по проблемам исследования, массовой практики и передового отечественного и зарубежного опыта, абстрагирование и конкретизация, сравнение, обобщение, моделирование результатов.

Практическая значимость полученных в работе результатов заключается в улучшении навыков разработки информационных систем для решения прикладных задач.

Выпускная квалификационная работа состоит из введения, трех глав, заключения, списка использованных источников, трех приложений. В первой главе проводится анализ предметной области и постановка задачи на разработку. Во второй главе выполняется проектирование информационной системы, определяются требования к аппаратным частям системы. В третьей главе приводятся результаты разработки, определяется экономическая эффективность от внедрения системы.

Глава 1 Функциональное моделирование предметной области

1.1 Техничко-экономическая характеристика учета операций кредитования в микрофинансовых организациях

Деятельность микрофинансовых организаций (МФО) заключается в оказании финансовых услуг предпринимателям и физическим лицам. В отличие от банков, МФО специализируются на небольших суммах (от 100 рублей), а также предъявляет меньше требований к заемщикам, не требуя от них иметь определенный уровень дохода и подтверждать его документами.

Цели функционирования микрофинансовых организаций [6], [18]:

- поддержка малого бизнеса;
- выдача микрозаймов населению.

Необходимость появления МФО вызвана тем, что банки не заинтересованы в обеспечении кредитной поддержки населения и предпринимателей, которые:

- находятся вне зоны покрытия банковскими отделениями, как правило, это небольшие населенные пункты, где численность населения варьируется в пределах нескольких сотен человек;
- желают получить сумму, не превышающую несколько десятков тысяч рублей, тогда как банки заинтересованы в выдаче кредита на сумму от 500 тысяч рублей;
- имеют плохую кредитную историю с задолженностями по кредитам.

Займы, выдаваемые МФО, имеют следующие характеристики:

- выдаются под определенный процент;
- должны быть возвращены в определенный срок;
- выдаются в течение короткого срока (решение о выдаче может быть принято за 10 минут);
- не требуют обязательного документального подтверждения величины и стабильности дохода.

Организационная структура микрофинансовой организации представлена на рисунке 1.

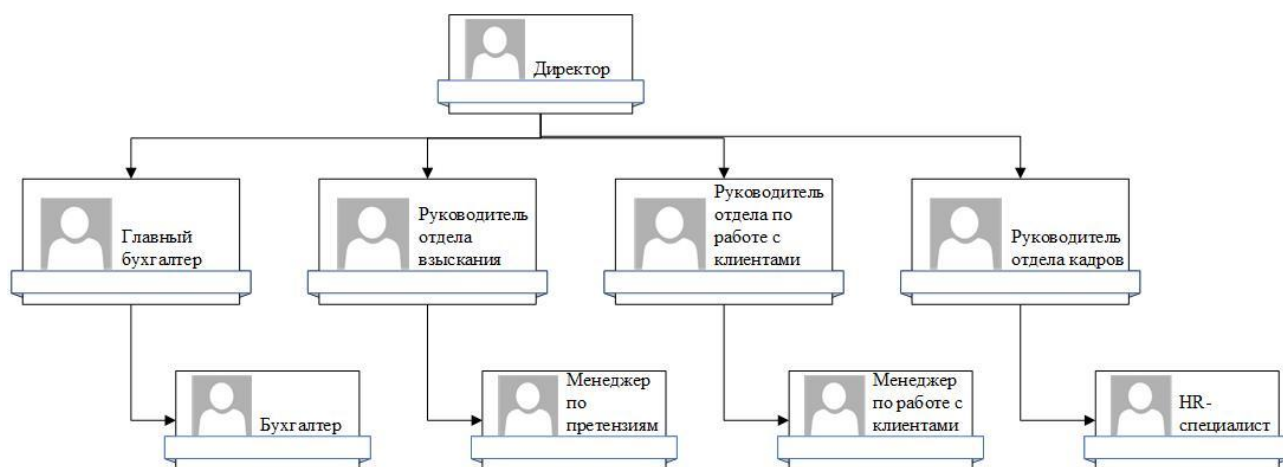


Рисунок 1 – Организационная структура МФО

В обязанности директора входит:

- организация деятельности компании, контроль над реализацией бизнес-процессов;
- взаимодействие с партнерами организации, арендодателями;
- контроль над своевременной подготовкой отчетности и отправкой в соответствующие надзорные органы.

В обязанности руководителя отдела кадров и его подчиненных входит подбор персонала, разработка кадровой политики.

Отдел по работе с клиентами рассматривает заявки на получение кредита (займа), консультирует клиентов по вопросам предоставления услуг, срокам, взысканиям.

Отдел по работе с взысканиями занимается клиентами-должниками. В задачи сотрудников отдела входит выяснение причин задолженности, оценка вероятности погашения долга, подготовка документов для обращения в суд, участие в судебных заседаниях.

В обязанности бухгалтерии входит управление финансовыми и нефинансовыми ресурсами организации, проведение расчетных операций, формирование финансовых документов и отчетов, оценка финансового состояния организации, отправка финансовых отчетов в контролирующие организации.

1.2 Концептуальное моделирование процесса по учету кредитных операций

1.2.1 Выбор технологии концептуального моделирования процесса по учету кредитных операций

Существует множество различных нотаций и методов для моделирования для бизнес-процессов организаций. Наиболее эффективными считаются следующие [3], [7]:

- Business Process Model and Notation (BPMN);
- Event-Driven Process Chain (EPC);
- Integrated Computer Aided Manufacturing Definition (IDEF0).

Нотация BPMN позволяет показать ход выполнения бизнес-процесса. На диаграмме могут быть отображены документы, события, действия, условия, ветвление. Для событий и шлюзов ветвления имеется большое количество обозначений, позволяющих продемонстрировать альтернативные, параллельные асинхронные действия. Достоинством нотации является подробное описание бизнес-процесса, возможность отображения альтернативных путей при решении задач. К недостаткам следует отнести сложность восприятия диаграмм при большом количестве элементов.

EPC основана на событиях. Нотация предназначена для построения схемы выполнения бизнес-процессов любого типа. При этом, каждому действию предшествует событие, после завершения действия также происходит событие. Достоинства нотации: подробное описание процесса, большое количество элементов для обозначения участников действий,

аппаратных и программных систем. Основным недостатком является большое количество тавтологических событий, например: действие «Сохранить документ» завершается событием «Договор сохранен». Не отображать данное событие нельзя, так как это нарушит требования нотации [11].

IDEFO позволяет строить иерархические модели процессов с различной степенью детализации операций. На верхнем уровне иерархии процесс отображен как единое целое, также на диаграмме показано внешнее окружение процесса: входные и выходные данные, управление и механизмы. Низкоуровневая детализация затрудняется тем, что IDEFO не содержит элементов для ветвления потоков. Часто для диаграмм на низших слоях иерархической структуры используются родственные нотации, позволяющие отобразить нелинейное движение процесса [7].

Для построения моделей бизнес-процесса по учету кредитных операций выбрана нотация bpmn, так как она позволяет отобразить алгоритм работы процесса со значительной детализацией, а также показать альтернативные варианты действий.

1.2.2 Моделирование процесса по учету кредитных операций для микрофинансовой организации для постановки задачи автоматизированного варианта решения

Автоматизация бизнес-процесса любого типа состоит из следующих этапов [7], [11]:

- изучение алгоритма работы процесса;
- построение модели текущей реализации процесса (модель «Как есть»);
- анализ построенной модели с целью поиска неоптимальных и избыточных действий;
- построение оптимальной модели процесса с использованием инструментов его автоматизации.

Основная цель автоматизации любой деятельности заключается в повышении ее эффективности. Повышение эффективности бизнес-процессов позволяет сокращать затраты на реализацию процессов, увеличивать прибыль организации.

При реализации бизнес-процесса могут быть задействованы несколько подразделений организации. Каждое подразделение решает свои задачи в рамках автоматизируемого процесса. Данный факт необходимо учитывать при выборе инструментов автоматизации. Выбранный инструмент должен обеспечивать одновременную работу сотрудников из разных подразделений с одним и тем же набором данных без нарушения целостности указанных данных.

Обработкой кредитных операций занимаются следующие подразделения:

- отдел взыскания;
- отдел по работе с клиентами.

Отдел взыскания включается в работу при наличии задолженностей у клиентов. Если задолженности отсутствуют, то всю работу с клиентами выполняет отдел по работе с клиентами. Сотрудники обоих подразделений должны иметь доступ к определенным данным по клиентам и кредитным операциям.

1.2.3 Разработка и анализ модели бизнес-процесса «Как есть»

Модель реализации бизнес-процесса по учету кредитных операций без автоматизации показана на рисунке 2. При неавтоматизированной реализации для сбора, хранения и обработки данных используются офисные приложения по работе с текстовыми документами и электронными таблицами. Данные хранятся в файлах, на бумажных носителях.

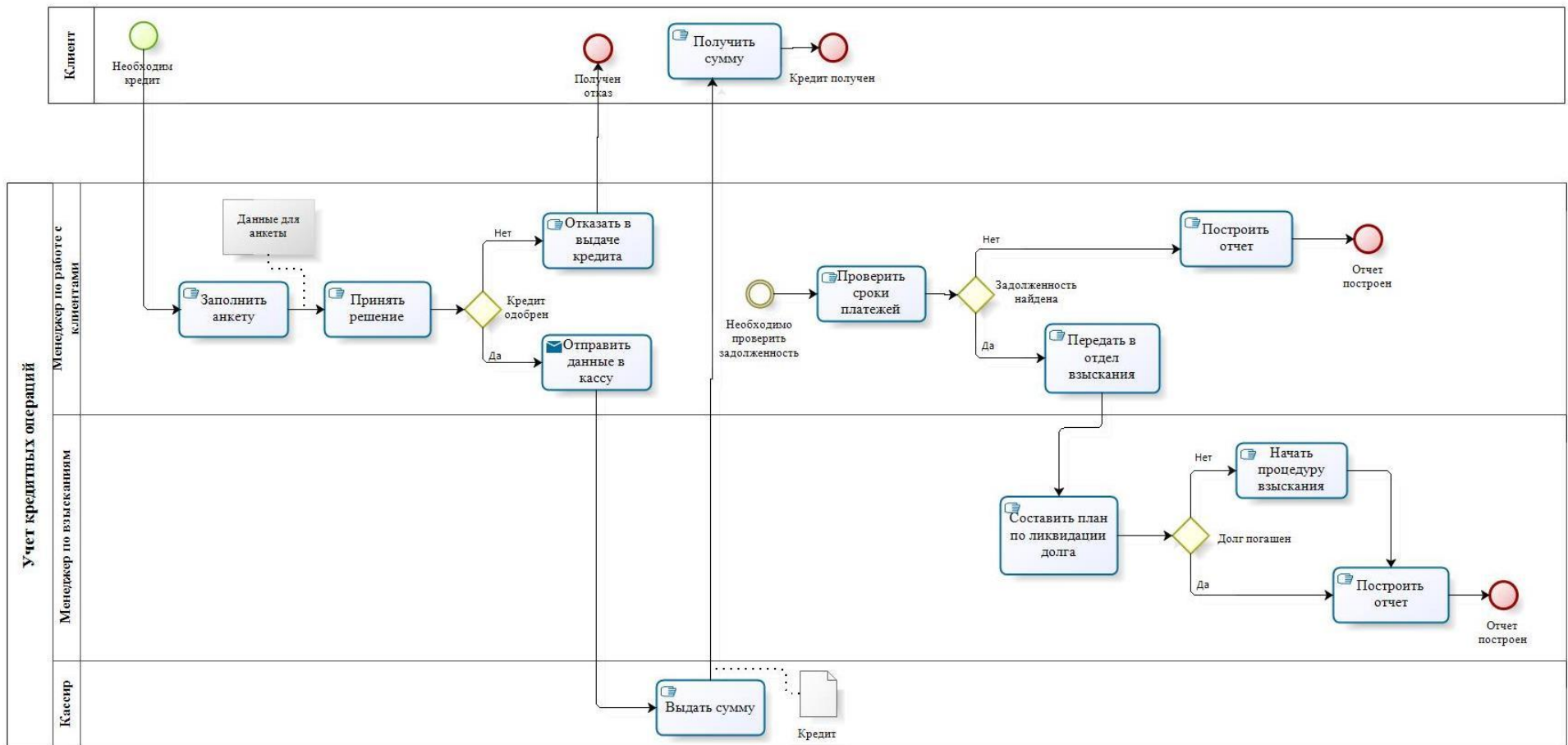


Рисунок 2 – Модель бизнес-процесса по учету кредитных операций «Как есть»

Неавтоматизированный процесс по учету кредитных операций имеет следующие недостатки:

- низкая производительность вычислительных задач;
- дублирование информации при отсутствии единого централизованного хранилища данных;
- сложность расчета показателей, характеризующих деятельность микрофинансовой организации, по причине большого объема данных;
- низкая оперативность обработки данных по кредитным операциям, следствием которой является принятие неверных или несвоевременных управленческих решений;
- неоптимальное хранение и сбор информации по клиентам и кредитным операциям;
- слабый уровень защиты данных по кредитным операциям, а также данных клиентов.

Для устранения указанных недостатков необходимо провести оптимизацию исходной версии реализации процесса.

1.2.4 Обоснование необходимости автоматизированного варианта решения и формирование требований к новой технологии

Скорость обработки данных по кредитным операциям, а также качество получаемых результатов существенно зависят от применяемых технологий и инструментов. Качество управленческих решений, принимаемых руководством микрофинансовой организации, также зависит от качества и своевременности результатов по обработке информации о кредитных операциях.

Таким образом, чем быстрее руководство будет получать отчетность, точно характеризующую деятельность микрофинансовой организации, тем быстрее будут приниматься управленческие решения, направленные на повышение благосостояния компании.

Получение качественных и своевременных результатов невозможно без использования современных информационных технологий, позволяющих оперативно и безошибочно обрабатывать большие объемы данных.

Автоматизация также позволит повысить эффективность выполнения рутинных операций по заполнению анкет за счет визуального интерфейса, подсказок, возможности выбора значения из списка и т.д. Помимо увеличения скорости заполнения анкет, автоматизация позволит сократить количество ошибок, возникающих по причине человеческого фактора, при внесении данных.

Основные требования к автоматизированной системе:

- создание, редактирование заявки на получение кредита;
- расчет графика платежей в случае одобрения заявки;
- оценка надежности заемщика по предоставленным данным и требуемой сумме;
- поиск данных по кредитным операциям, клиентам;
- построение отчетов по данным с выбором временного периода.

Требования к качеству системы: надежность, масштабируемость, обновляемость.

Распределение функций между пользователем и системой:

- пользователь заполняет электронные формы, выбирает необходимые действия;
- система выполняет проверку корректности вводимых данных, вычисления, поиск и другую обработку данных с последующим представлением результатов обработки пользователю.

Безопасность данных должна обеспечиваться за счет:

- механизмов авторизации и аутентификации пользователей;
- разграничения прав доступа к данным;
- контроля над целостностью и непротиворечивостью хранимых данных.

Показателем успешности применения программного обеспечения будет являться эффективное решение сотрудниками организации рутинных задач с помощью системы. В таком случае проектируемая система будет востребована в организации.

1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям

1.3.1 Определение критериев анализа

Существует два способа приобретения информационной системы для автоматизации любого бизнес-процесса:

- настройка готового решения;
- разработка системы с нуля.

Для выбора наиболее оптимального варианта необходимо составить список критериев, выполнение которых позволит приобрести систему, решающую все необходимые задачи и обеспечивающую необходимый уровень безопасности данных. В случае выбора системы для автоматизации учета кредитных операций целесообразно использовать следующие критерии: поддержка многопользовательской работы с базой данных, требовательность к ресурсам, отсутствие невостребованных функций, стоимость системы.

Далее необходимо провести анализ готовых решений на предмет соответствия указанным критериям. Степень соответствия критерию будет обозначена следующим образом:

- «+» – полное соответствие критерию;
- «-» – полное не соответствие критерию;
- «+/-» – частичное соответствие;
- «-/+» – система практически не соответствует критерию.

На основе проведенного анализа необходимо сделать вывод о целесообразности использования готового решения, либо собственной разработки информационной системы.

1.3.2 Сравнительная характеристика существующих разработок

Решение от компании 1С – 1С: Управление микрофинансовой организацией и кредитным потребительским кооперативом – предназначено для автоматизации деятельности микрофинансовых организаций. Основные возможности системы: поддержка деятельности по привлечению и предоставлению финансовых средств, ведение бухгалтерского и налогового учета, построение стандартных бухгалтерских отчетов.

При необходимости, можно использовать дополнительные модули для фондов и ломбардов. Функции решения: расчет процентного дохода, ведение учета по отдельным подразделениям в соответствии с требованиями регулирующих организаций, разработка финансовых продуктов, ведение базы клиентов, расчет графиков выплат, оценка надежности клиента, анализ отклонений от плана погашения задолженности, сервисные функции.

Минимальная стоимость решения составляет 29000р.

Решение от компании HES реализует следующие функции: полная автоматизация кредитных процессов, встроенная CRM-система, интеграция с БКИ, платежными агрегаторами, 1С, настройка собственного колл-центра, скоринговый алгоритм для оценки платежеспособности клиента.

Система обладает высокими требованиями к ресурсам, стоимость определяется индивидуально для каждой компании.

Онлайн-решение от компании Спутник – Credit Conveyer – предназначено для автоматизации различных бизнес-процессов микрофинансовых организаций.

Функции системы: модуль CRM, управление займами, построение аналитических отчетов, интеграция с 1С, большое количество ролей сотрудников.

Результаты сравнения рассмотренных программных решений приведены в таблице 1.

Таблица 1 – Сравнение готовых решений

	1С	НЭС	Спутник
Поддержка многопользовательской работы с базой данных	+	+	+
Требовательность к ресурсам	-	-/+	+/-
Отсутствие невостребованных функций	-/+	-/+	-/+
Стоимость системы	+/-	-	+/-

Как видно из таблицы 1, ни одно из рассмотренных решений не соответствует всем выбранным критериям. Следовательно, целесообразным выбором является собственная разработка системы по учету кредитных операций.

1.4 Постановка задачи на разработку системы

Проектируемая информационная система предназначена для ведения учета кредитных операций в микрофинансовой организации. Информационная система должна иметь клиент-серверную архитектуру с поддержкой многопользовательского режима. Требования к программе: программа должна иметь интуитивно-понятный интерфейс, все элементы управления должны иметь стандартное назначение, доступ к данным должен быть разграничен в зависимости от типа учетной записи пользователя.

Функции системы: формирование заявки на получение кредита, редактирование заявки, оценка надежности потенциального заемщика, формирование и корректировка графика выплат, построение отчетов по клиентам, кредитным операциям, проверка нарушения графика выплат.

1.5 Разработка модели бизнес-процесса «Как должно быть»

Схема автоматизированного бизнес-процесса учета кредитных операций показана на рисунке 3.

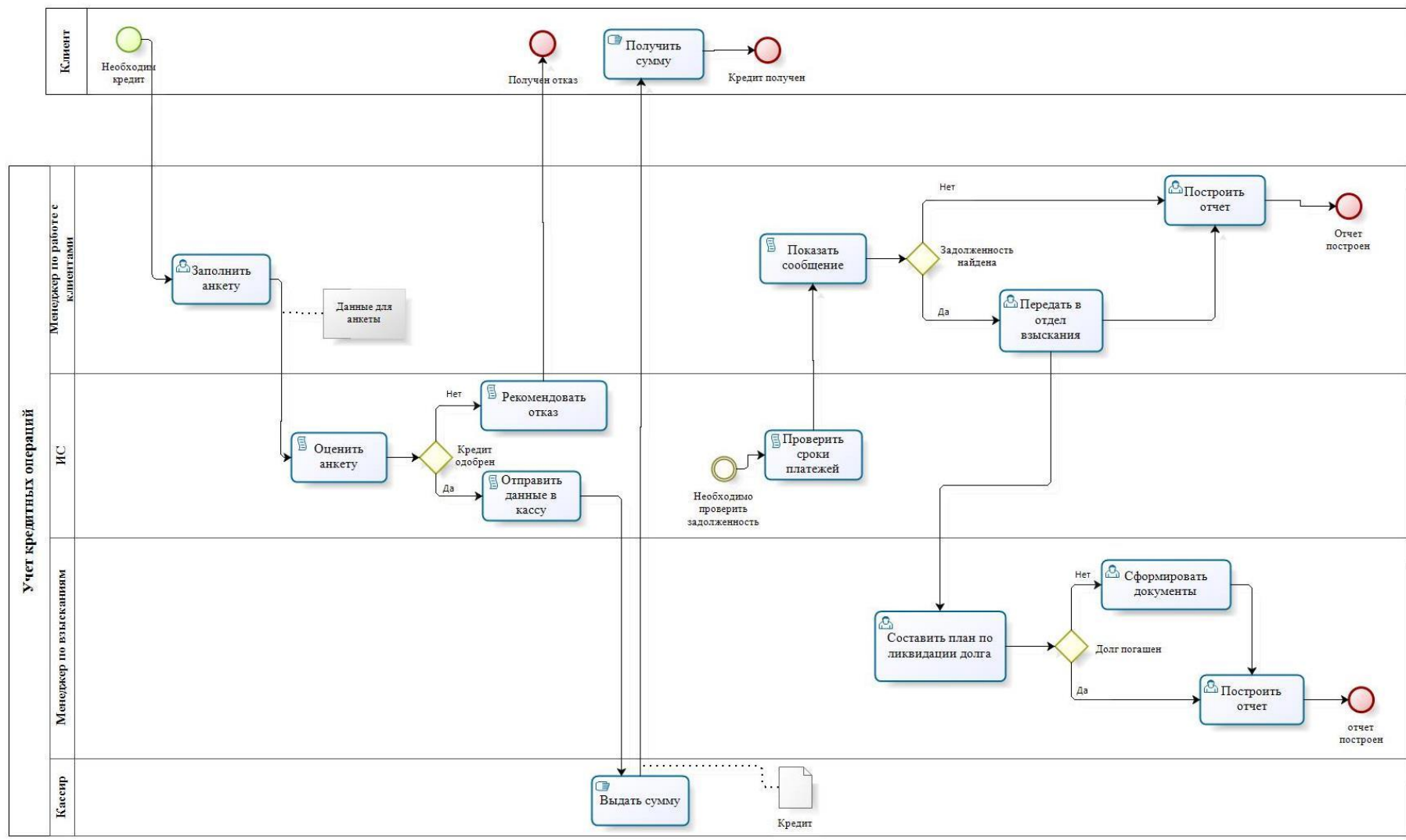


Рисунок 3– Схема бизнес-процесса по учету кредитных операций «Как должно быть»

Автоматизация процесса по учету кредитных операций позволит:

- сделать часть задач (оценка анкеты, проверка своевременности выплат и т.д.) полностью автоматическими, что позволит ускорить время их выполнения, а также исключить вероятность ошибки по причине человеческого фактора;
- задачи, в которых пользователю необходимо вводить или выбирать исходные данные, будут выполняться через визуальный интерфейс системы, вводимые данные будут проверяться системой на корректность;
- повысить эффективность выполнения процесса.

Выводы по главе 1

В первой главе представлена характеристика деятельности микрофинансовой организации, проведено концептуальное моделирование бизнес-процесса по учету кредитных операций: построены модели «Как есть» и «Как должно быть», проведен обзор готовых решений по автоматизации исследуемого бизнес-процесса, обоснована целесообразность собственной разработки системы, а также необходимость автоматизации исследуемого бизнес-процесса.

Глава 2 Логическое проектирование системы по учету кредитных операций

2.1 Выбор технологии логического моделирования информационной системы

Для построения модели информационной системы по учету кредитных операций выбран язык объектно-ориентированного моделирования процессов и систем UML (Unified Modeling Language). Диаграммы, построенные с помощью UML, позволяют отобразить структуру или функции приложения целиком, либо одного из его модулей.

Модель системы, построенная с помощью UML, состоит из набора диаграмм, каждая из которых характеризует систему целиком, либо ее часть. UML-диаграммы подразделяются на две группы [2], [5]:

- структурные – отображают компоненты системы и их взаимосвязь;
- поведенческие – отображают функциональность информационной системы.

Язык UML позволяет построить модель любого процесса или системы, вне зависимости от их уровня сложности. UML представляет собой общий язык для бизнес-аналитиков, архитекторов программного обеспечения и разработчиков, используемый для описания, спецификации, проектирования и документирования существующих или новых бизнес-процессов, структуры и поведения артефактов программных систем.

UML может быть использован в различных областях: в банковском деле, финансах, Интернете, аэрокосмической отрасли, здравоохранении и т. д. Его можно использовать со всеми основными методами разработки объектного и компонентного программного обеспечения и для различных платформ реализации.

UML не зависит от процесса разработки программного обеспечения и может применяться в контексте различных процессов. Однако, наиболее

оптимальным является использование UML-моделей при применении итеративных и инкрементных процессов разработки.

2.2 Логическая модель системы по учету кредитных операций и ее описание

Диаграммы вариантов использования позволяют отобразить функциональные возможности проектируемого приложения. Основными элементами диаграммы являются:

- Актор – внешняя система или пользователь, которому доступны функции системы;
- Вариант использования – функция проектируемого программного продукта.

В приложении для учета кредитных операций будут задействованы две роли пользователей: Менеджер по работе с клиентами и Менеджер по розысканиям. Диаграмма вариантов использования для менеджера по работе с клиентами показана на рисунке 4.

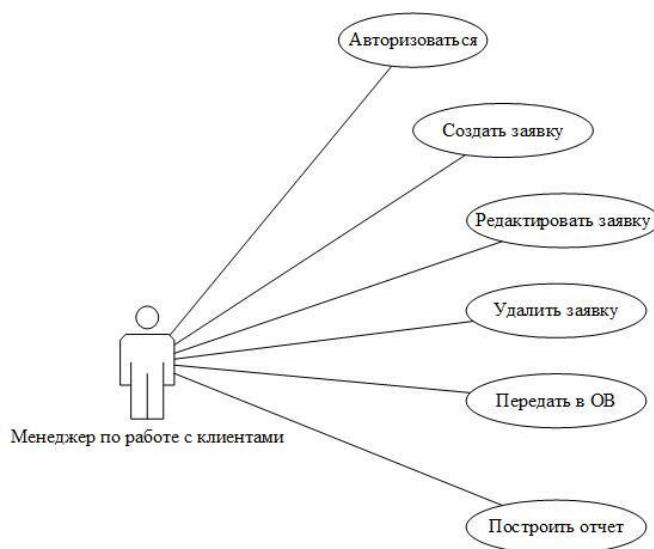


Рисунок 4 – Диаграмма вариантов использования (менеджер по работе с клиентами)

Как видно из рисунка 4, менеджеру доступны авторизация, обработка данных по заявке на кредит, передача данных о задолженности в отдел взыскания, построение отчетов различного типа.

На рисунке 5 представлена диаграмма вариантов использования для менеджера по взысканиям.



Рисунок 5 – Диаграмма вариантов использования (менеджер по взысканиям)

Как видно из рисунка, пользователю доступны авторизация в системе, построение документов (для обращения в суд в случае, если кредит не выплачен несмотря на изменение плана выплат) и отчетов, формирование нового плана выплат для должника.

На рисунке 6 показана диаграмма пакетов, отображающая общую структуру приложения.

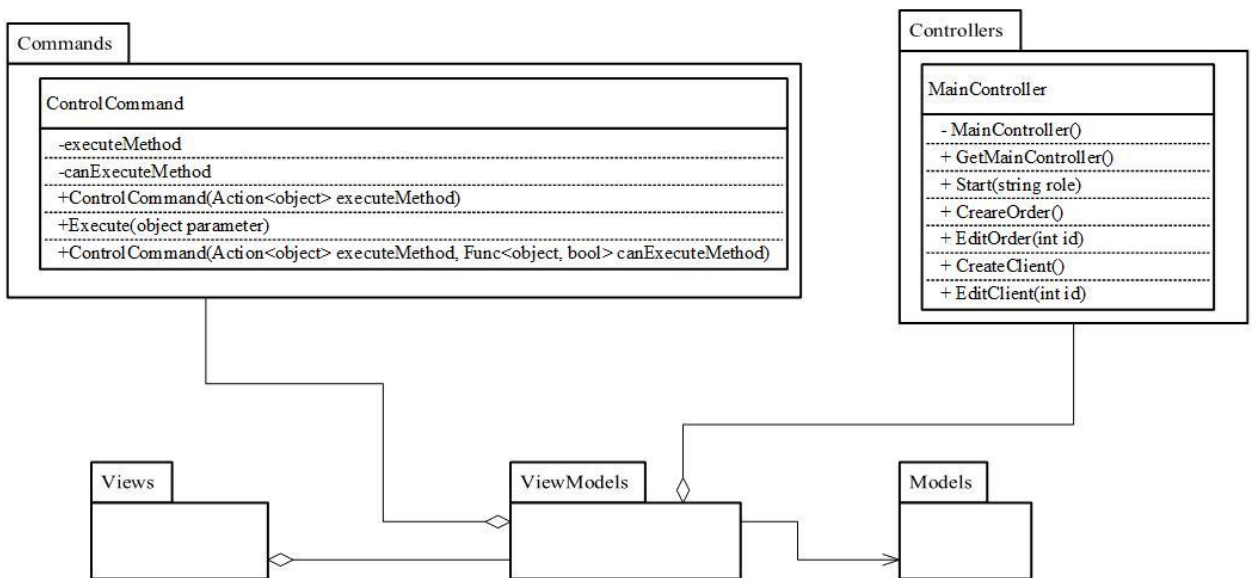


Рисунок 6 – Диаграмма пакетов

При построении структуры приложения выбран паттерн MVVM (Model View ViewModel). Данный шаблон позволяет разделить данные, интерфейс и логику по обработке данных на отдельные модули так, что модификация одного модуля не требует изменения остальных [1], [4]. На рисунке 6 представлены следующие пакеты (модули):

- Model (Модель) – пакет содержит классы, на основе которых создаются реляционные отношения базы данных.
- View (Представление) – классы, представляющие интерфейс приложения: оконные формы с кнопками, текстовыми полями, таблицами и т.д.
- ViewModel (Модель представления) – классы, содержащие логику по обработке данных из Model, а также по обработке взаимодействия пользователя с интерфейсом (клик по кнопкам, вызов пунктов меню и т.д.).
- Commands – содержит классы для реализации механизма команд, предназначенного для связи интерфейса и логики по обработке данных, не имеет прямого отношения к шаблону MVVM.

- **Controllers** – содержит классы, обеспечивающие вызов экранных форм, также не имеет прямого отношения к шаблону MVVM.

На рисунке 7 показана диаграмма классов [8], [12], [14], входящих в пакет Models:

- **Client** – клиент микрофинансовой организации, обратившийся за получением займа;
- **User** – пользователь проектируемого приложения, сотрудник микрофинансовой организации;
- **Order** – заявка от клиента на получение кредита;
- **Document** – документы, относящиеся к заявке;
- **Insurance** – страховка для займа, оформляется по желанию клиента;
- **PayPlan** – план-график выплат по займу;
- **MContext** – объектно-ориентированный аналог реляционной базы данных.

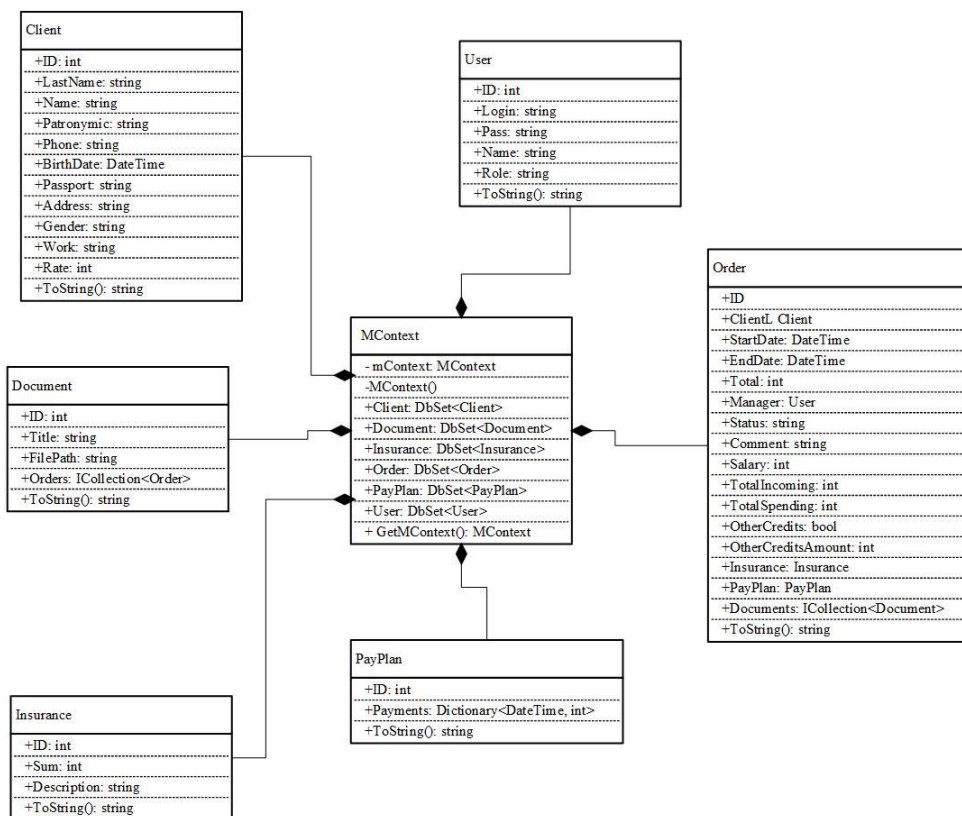


Рисунок 7 3 – Диаграмма классов для пакета Models

На рисунке 8 приведена диаграмма классов для пакета ViewModels:

- BaseViewModel – родительский класс для всех моделей представления, содержит поля и методы, общие для всех моделей представления.
- StartViewModel – модель представления, обеспечивающая работу функции авторизации: поиск в базе логина и пароля, открытие соответствующего личного кабинета, либо отображение ошибки авторизации.
- ClientManagerViewModel – модель представления, реализующая функции личного кабинета менеджера по работе с клиентами.
- ReturnManagerViewModel – модель представления, реализующая функции личного кабинета менеджера по взысканиям.

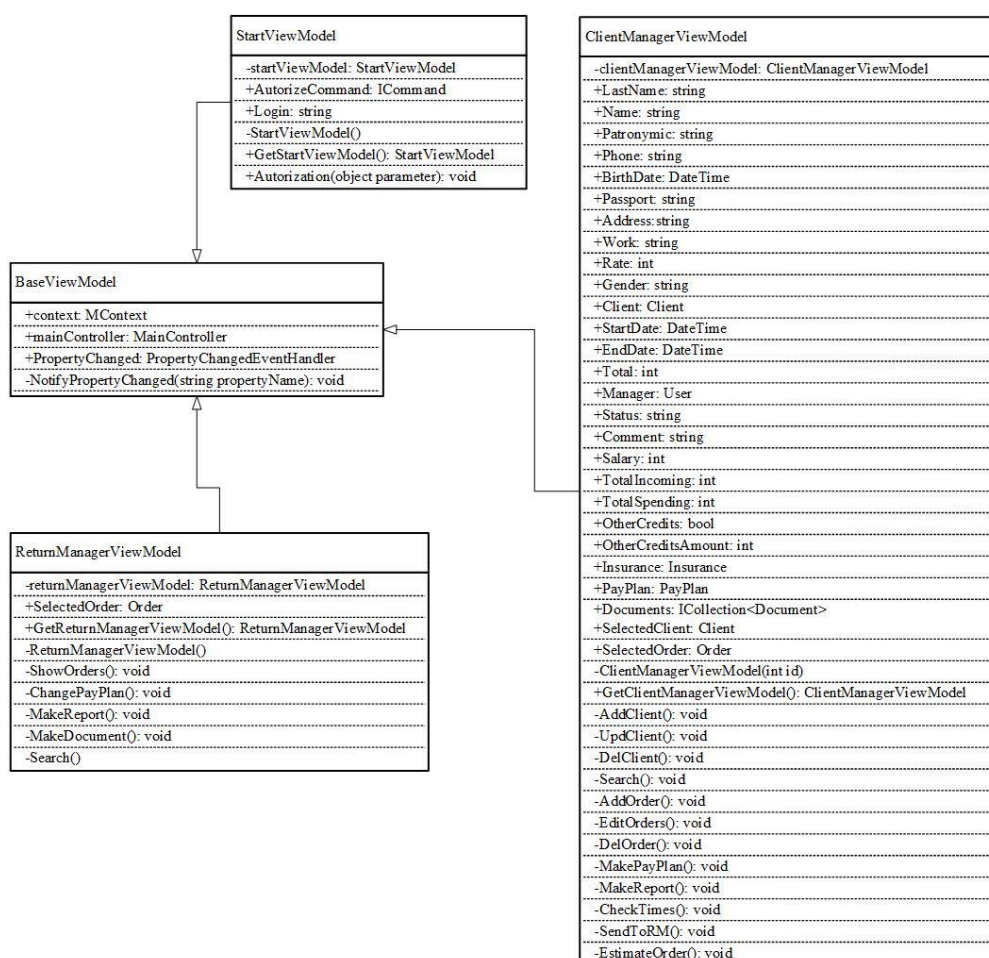


Рисунок 8 – Диаграмма классов для пакета ViewModels

На рисунке 9 показана диаграмма классов для пакета Views:

- LoginWindow – форма авторизации.
- ClientManagerWindow – личный кабинет менеджера по работе с клиентами.
- ReturnManagerWindow – личный кабинет менеджера по взысканиям.
- AddOrderWindow – форма для добавления или редактирования данных о клиенте.
- AddClientWindow – форма для создания новой заявки, либо изменения уже созданной заявки.

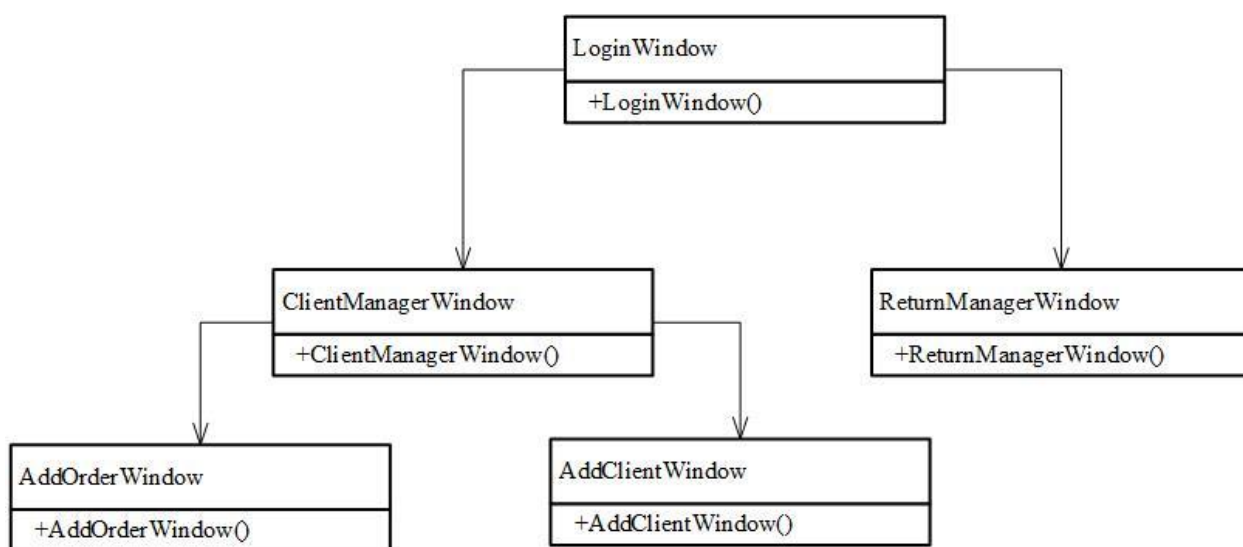


Рисунок 9 – Диаграмма классов для пакета Views

Классы пакета Views – оконные формы проектируемой системы, ее интерфейс. При запуске приложения открывается окно с формой авторизации, далее, в зависимости от типа учетной записи пользователя, выполняется переход на окно личного кабинета.

2.3 Информационное обеспечение системы

2.3.1 Используемые классификаторы и системы кодирования

Обозначения для кодируемых объектов приведены в таблице 2.

Таблица 2 – Используемые классификаторы и коды

Название объекта кодируемого множества	Значимость кода	Система кодирования	Вид классификатора	Пример
Код заявки	10	Р	Л	05052022/1
Код клиента	8	П	Л	17895489
Код сотрудника	2	П	Л	23
Код документа	10	П	Л	1009768943
Код страховки	12	Р	Л	05052022/1/1
Код плана выплат	12	Р	Л	05052022/1/2

В таблице 2 и последующих таблицах использованы следующие обозначения для систем кодирования:

- Р – разрядная, т.е., классификация выполняется с использованием определенных разрядов, в данном случае, используется дата и порядковый номер для кода заявки, а также код заявки и порядковый номер для страховки и плана;
- П – порядковая, т.е. каждому объекту ставится в соответствие порядковый номер.

2.3.2 Характеристика нормативно-справочной информации

При реализации процесса по учету кредитных операций используются следующие первичные документы:

- анкета клиента;
- заявка на кредит.

Анкета клиента содержит следующие реквизиты: ФИО, телефон, дата рождения, паспортные данные, адрес, пол, должность.

Реквизиты для заявки на кредит: клиент, процентная ставка, даты выдачи и погашения кредита, сумма кредита, зарплата и общий доход клиента, расходы за месяц, наличие и количество других кредитов.

Структура реляционного отношения базы данных, в котором хранятся анкеты, приведена в таблице 3.

Таблица 3 – Структура анкеты

Название поля	Идентификатор	Тип	Значность
ФИО	name	C	300
телефон	phone	C	20
дата рождения	birthdate	D	20
паспортные данные	passport	C	15
адрес	address	C	300
пол	gender	C	1
должность	work	C	100

Структура реляционного отношения базы данных, в котором хранятся заявки, приведена в таблице 4.

Таблица 4 – Структура заявки

Название поля	Идентификатор	Тип	Значность
клиент	client	C	300
процентная ставка	percent	N	2
дата выдачи кредита	startdate	D	20
дата погашения кредита	enddate	D	20
сумма кредита	total	N	10
зарплата клиента	salary	N	10
общий доход клиента	totalincoming	N	10
расходы за месяц	totalspending	N	10
наличие других кредитов	othercredits	N	1
количество других кредитов	othercreditsamount	N	2

В таблицах используются следующие обозначения: C – символьные данные, N – числовые данные, D – дата и время.

2.3.3 Характеристика выходной информации

Выходной информацией являются:

- план-график платежей по кредиту;

– отчеты по клиентам и кредитам.

Реквизиты плана-графика приведены в таблице 5.

Таблица 5– Структура плана-графика

Название поля	Идентификатор	Тип	Значность
Дата платежа	paydate	D	20
Сумма платежа	paysum	N	10
Номер заявки	order	C	10

Реквизиты отчета по клиентам приведены в таблице 6.

Таблица 6 – Структура отчета по клиентам

Название поля	Идентификатор	Тип	Значность
ФИО	fio	C	300
Общее количество займов	total	N	3
Количество просроченных займов	overdue	N	3
Количество своевременно выплаченных займов	intime	N	3
Количество выплачиваемых займов	paying	N	3
Дата построения отчета	createdate	D	20
Временной период	period	D	40

Реквизиты отчета по кредитам приведены в таблице 7.

Таблица 7 – Структура отчета по кредитам

Название поля	Идентификатор	Тип	Значность
Временной период	fio	C	300
Общая сумма займов	total	N	10
Количество клиентов	ctotal	N	5
Количество своевременно выплаченных займов	intime	N	5
Количество просроченных займов	overdue	N	5
Дата построения отчета	createdate	D	20

2.4 Проектирование базы данных информационной системы по учету кредитных операций

2.4.1 Выбор технологии проектирования базы данных для информационной системы

Для проектирования базы данных информационной системы учета кредитных операций использованы нотация П. Чена (концептуальная схема) и нотация IDEF1X для логической модели.

Нотация Чена позволяет отобразить основные сущности предметной области и связи между ними. Модель не зависит от предметной области, типа проектируемой базы данных, инструментов проектирования и разработки. Таким образом, для начального этапа проектирования базы данных нотация Чена является одним из оптимальных вариантов [9], [10], [13].

Нотация моделирования IDEF1X (IDEF1 eXtended) предназначена для семантического моделирования данных на основе концепции «Сущность-Связь». Основными составляющими модели являются:

- Сущности: абстрактные или реальные объекты предметной области, которые обладают общими характеристиками;
- Отношения: связи между сущностями, количество и вид связей зависит от предметной области, связи имеют кратность, показывающую количество сущностей, участвующих в связи с каждой из сторон;
- Атрибуты: характеристики сущностей, каждая сущность имеет собственный набор значений для атрибутов.

Нотация IDEF1X [10] является одним из оптимальных вариантов для построения логической модели, так как не зависит от средств реализации базы данных. Для понимания схемы не нужны специфические знания. Модель, построенная с использованием IDEF1X, отображает структуру сущностей и их взаимосвязи, что является достаточным для разработки базы данных.

2.4.2 Разработка концептуальной модели данных для информационной системы

Концептуальная модель базы данных приложения по учету кредитных операций показана на рисунке 10.

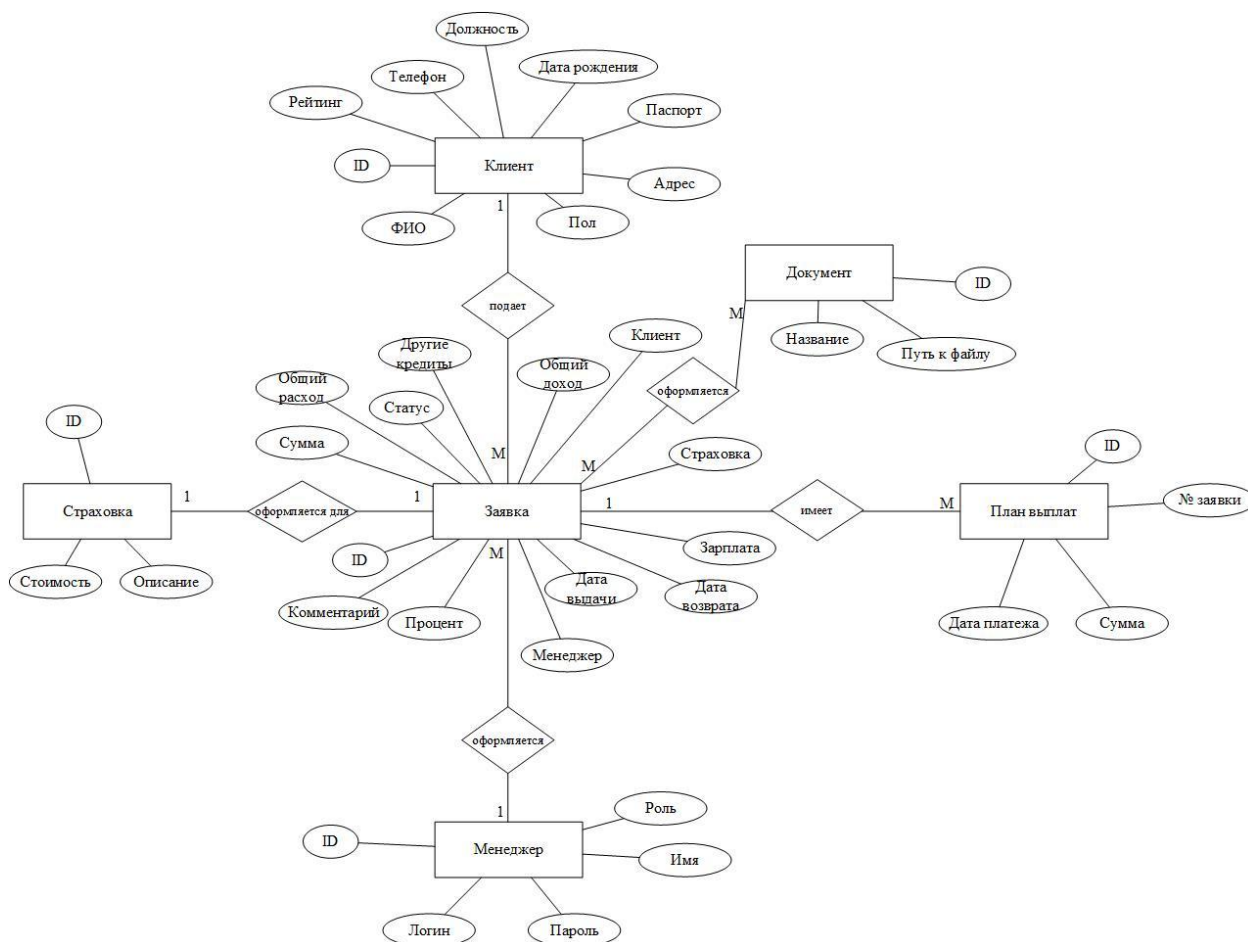


Рисунок 10 – Концептуальная модель базы данных

На рисунке 10 представлены следующие сущности:

- Клиент – клиент микрофинансовой организации, обратившейся за кредитом;
- Документ – документ, оформляемый к заявке по кредиту при необходимости;
- Страховка – оформляется для заявки при необходимости или по желанию клиента;

- Заявка – заявка клиента на получение кредита;
- План выплат – список дат и сумм, которые должны быть уплачены в указанные даты;
- Менеджер – сотрудник микрофинансовой организации, который оформляет заявку, а затем контролирует своевременность выплат по кредиту.

2.4.3 Обоснование вида логической модели

Логическая модель базы данных строится на основе концептуальной модели. Логическая модель не зависит от реализации базы данных, но на этапе логического проектирования необходимо выбрать модель данных для проектируемой базы. Основными моделями данных являются: иерархическая, сетевая, реляционная [9], [10].

Иерархическая модель – представляет собой структуру данных, в которой каждый новый элемент связан только с одним своим предком. Как правило, база имеет древовидную структуру. Модель имеет ряд ограничений, которые затрудняют ее активное использование в проектах.

Сетевая модель – расширенная иерархическая модель, в которой увеличено количество связей. На практике, у подобной модели будет быстро расти сложность представления данных, что также затрудняет ее применимость на практике.

Реляционная модель – согласно данной модели, вся информация хранится в двумерных таблицах (реляционных отношениях) с заранее определенной структурой. Каждое отношение хранит информацию об одном объекте предметной области. Получение всей информации о предметной области осуществляется за счет связей между отношениями. Реляционная модель является наиболее востребованной при создании хранилищ данных на основе СУБД.

Таким образом, реляционная модель данных является наиболее оптимальным вариантом для построения базы данных приложения по учету кредитных операций.

2.4.4 Разработка логической модели данных системы

Логическая модель базы данных показана на рисунке 11.

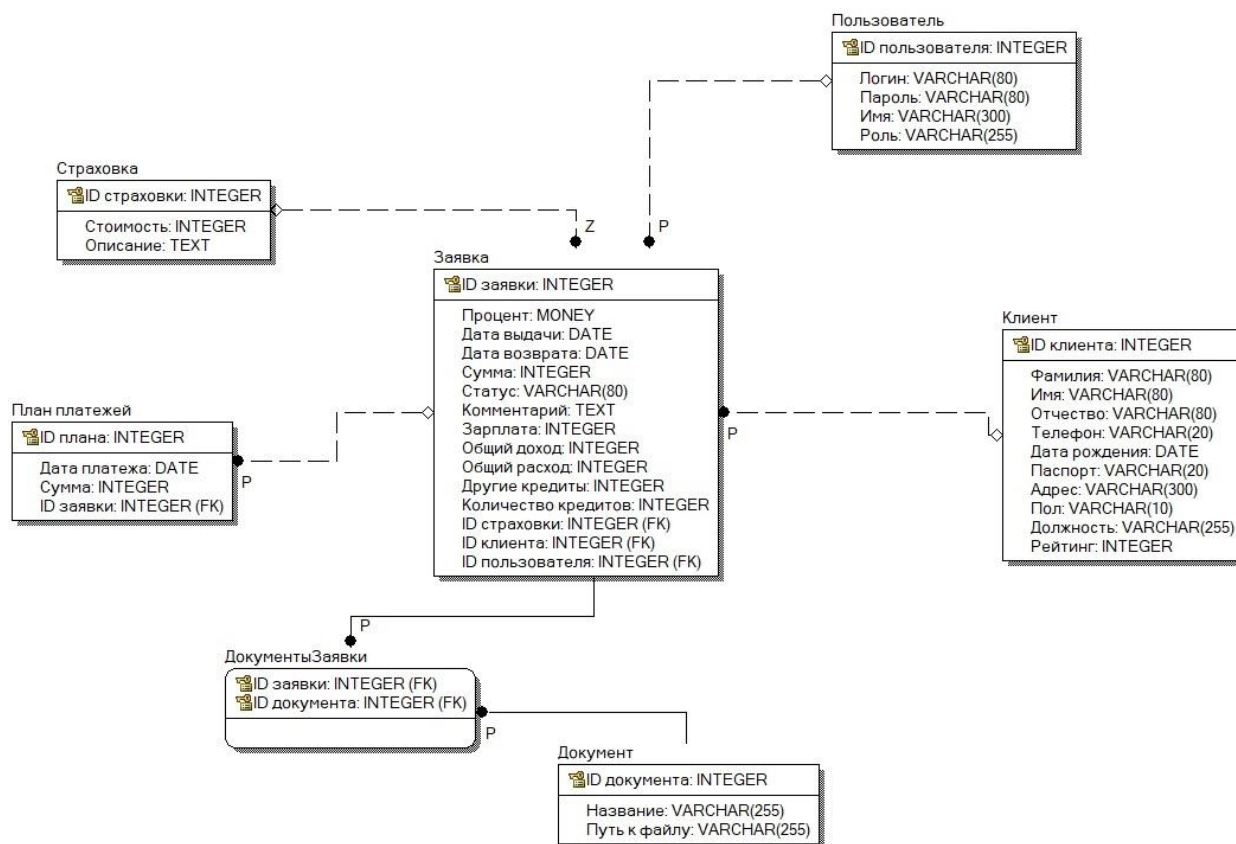


Рисунок 11 – Логическая модель данных

Разработанные реляционные отношения находятся в первой нормальной форме (1НФ), так как значения всех ячеек являются атомарными. Реляционные отношения соответствуют второй нормальной форме (2НФ), так как они находятся в 1НФ и все неключевые атрибуты для каждого отношения неприводимо зависят от ключевого атрибута. Реляционные отношения находятся в третьей нормальной форме (3НФ), так как они находятся в 2НФ и транзитивные зависимости между неключевыми атрибутами отсутствуют.

2.5 Требования к аппаратно-программному обеспечению системы по учету кредитных операций

Для бесперебойной работы проектируемой информационной системы необходимо наличие оборудования, соответствующего указанным ниже минимальным требованиям.

Требования к серверу баз данных:

- процессор, с частотой от 2,5 ГГц, количество ядер процессора – от 4;
- оперативная память от 8 ГБ;
- количество жестких дисков от 2 шт, со скоростью вращения не менее 10000 оборотов в минуту.

Требования к клиентским компьютерам:

- частота процессора 1.6 ГГц, количество ядер – 2;
- размер оперативной памяти 4 Гб;
- жесткий диск размером от 512 Мб.

Выводы по главе 2

Во второй главе представлены результаты проектирования базы данных информационной системы по учету кредитных операций, а также интерфейса системы и логики по обработке данных. Представлены характеристики входных и выходных данных. Обоснован выбор методов и инструментов для проектирования системы.

Глава 3 Физическое проектирование информационной системы по учету кредитных операций

3.1 Выбор архитектуры информационной системы по учету кредитных операций

Хранение данных на сервере обеспечивает следующие преимущества:

- централизованное хранение данных;
- обеспечение защиты хранимых данных;
- распределенный доступ к серверу с нескольких рабочих станций;
- возможность разделения функций по обработке данных между клиентом и сервером.

Существуют следующие основные виды клиент-серверной архитектуры:

- файл-сервер;
- n-звенный клиент-сервер.

Архитектура «Файл-сервер» работает следующим образом: на выделенном компьютере располагаются данные, к выделенному компьютеру имеют доступ другие компьютеры сети (клиентские). Клиентские компьютеры могут скачивать данные с файлового сервера для последующей обработки. Недостатком данной архитектуры является передача избыточных объемов данных. Это связано с тем, что на файловом сервере отсутствует возможность обработки данных.

При использовании архитектуры «N-звенный клиент-сервер» в сети имеются несколько серверов: на одном хранятся данные, на остальных распределены приложения с функциями по обработке данных (серверы приложений). Обработка данных выполняется на серверах приложений, что позволяет снизить нагрузку на сеть и передавать клиентскому компьютеру результаты обработки данных, а не исходные данные. Недостатком данной

архитектуры является сложная структура сети, высокие затраты на оборудование, сложности в настройке и интеграции компонентов [15] – [17].

При реализации информационной системы для микрофинансовой организации выбрана архитектура «2-звенный клиент-сервер». На сервере баз данных расположены данные, клиентский компьютер через приложение отправляет на сервер запрос на обработку данных, данные обрабатываются средствами СУБД, на клиентский компьютер отправляются результаты обработки. Данное решение позволит снизить нагрузку на сеть, а также использовать оптимальную по сложности конфигурацию сети.

Для отображения аппаратной структуры проектируемой системы используется диаграмма развертывания. На рисунке 12 показана диаграмма развертывания для информационной системы по учету кредитных операций.

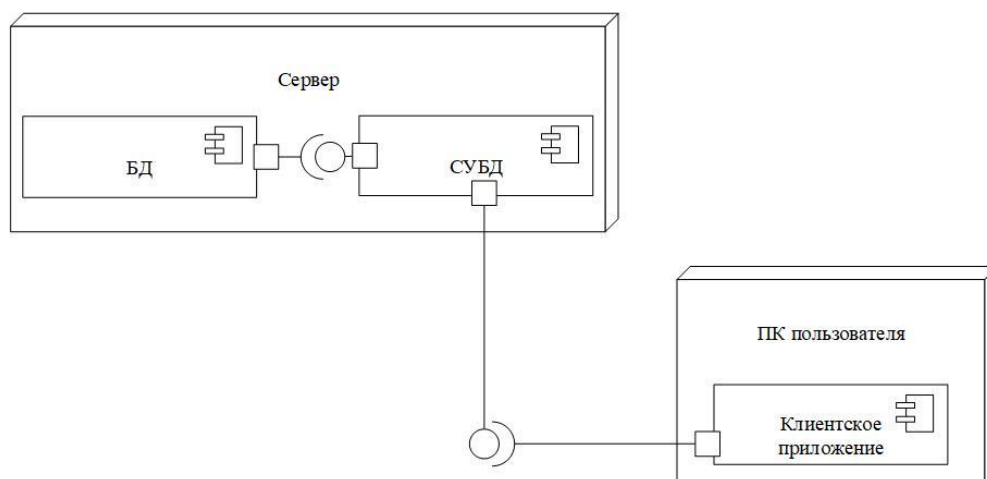


Рисунок 12 – Диаграмма развертывания

Количество клиентских компьютеров определяется количеством рабочих мест для сотрудников.

3.2 Выбор технологии разработки программного обеспечения информационной системы по учету кредитных операций

Существуют следующие технологии разработки программного обеспечения: каскадная, спиральная, RAD.

Каскадная модель (водопад): перед разработкой программного продукта происходит формулирование требований к системе, далее в ходе разработки, эти требования остаются неизменными. Каждый этап выполняется один раз, далее происходит составление документации и переход к следующему этапу [19] - [20].

При использовании спиральной модели на каждом этапе разработки проводится анализ созданных модулей и принимается решение о переходе на следующий этап или возвращении на предыдущий с целью доработки функционала системы.

После прохождения всех этапов модели получается некоторая промежуточная версия программного продукта. Таким образом, требования к системе могут меняться в ходе разработки, изменения можно вносить, не дожидаясь готовности очередной версии системы.

Одним из наиболее распространенных подходов к разработке программных продуктов, в рамках спиральной модели, является методология RAD (Rapid Application Development). Основные характеристики данной методологии [21] – [23]:

- небольшая команда разработчиков (до десяти человек);
- на разработку отводится от двух до шести месяцев;
- постепенное наращивание функционала разрабатываемой системы.

Жизненный цикл, согласно RAD, делится на следующие этапы:

- анализ и планирование требований – определение функций, которые будут реализованы на текущей итерации;
- проектирование системы – построение прототипа системы с использованием CASE-средств;

- построение системы – разработка кода;
- внедрение системы – развертывание системы на базе заказчика, обучение конечных пользователей работе с системой.

При разработке информационной системы для микрофинансовой организации выбрана методология RAD, так как данная методология эффективна при работе над небольшими проектами, разрабатываемыми под определенного заказчика. А также обладает гибкостью при разработке, позволяя создавать прототипы системы с определенным набором функциональных возможностей. Что позволяет контролировать соответствие наращиваемой функциональности требованиям, поставленным изначально.

Далее необходимо выбрать инструменты для разработки информационной системы. Для разработки выбран объектно-ориентированный язык C#, сравнение сред разработки, поддерживающих C#, приведено в таблице 8.

Таблица 8 – Сравнение сред разработки

	Наличие бесплатной версии	Функциональные возможности	Инструменты для ускорения разработки	Поддержка WPF
Visual Studio	+	+	+	+
Project Rider	-	+	+/-	+
Eclips	+	+	+/-	-
Visual Studio Code	+	+/-	+/-	-
Mono Develop	+	+	+/-	-

Сравнение проведено по следующим параметрам:

- наличие бесплатной версии – чем ниже цена программного продукта и его плагинов, тем меньше расходы на разработку;
- функциональные возможности – среда разработки должна обеспечивать пошаговую отладку и форматирование кода, построение интерфейса в визуальном и текстовом редакторах, создание тестов;

- инструменты для ускорения разработки – наличие комбинаций клавиш, при нажатии на которые создаются шаблоны свойств, методов и т.д. позволяет значительно ускорить процесс написания кода;
- поддержка WPF – поддержка современной и многофункциональной технологии разработки визуального интерфейса позволит создать современное и удобное для использования приложение.

Знаками «+», «-», «+/-» обозначена степень соответствия критерию. Project Rider является коммерческим программным обеспечением, следовательно, критерию «наличие бесплатной версии» она не соответствует. Eclipse, Visual Studio Code, Mono Develop не поддерживают WPF. Visual Studio обладает наиболее широкими функциональными возможностями.

Как видно из таблицы 8, наиболее оптимальным вариантом является Visual studio. Версия Community содержит все необходимые для создания разрабатываемой системы функциональные возможности.

3.3 Выбор системы управления базой данных для информационной системы по учету кредитных операций

СУБД Oracle рассчитана на средние и крупные программные системы, работающие с базами данных. Решения от Oracle обладают высокой степенью защиты данных, наиболее широкими функциональными возможностями. Решения от Oracle распространяются по бесплатной и коммерческой лицензии, но, бесплатные решения заметно ограничены в использовании. Базы данных Oracle работают со следующими семействами операционных систем: Windows, Linux, UNIX. Помимо СУБД, Oracle выпускает несколько видов клиентских приложений, позволяющих создавать полноценные информационные системы для организаций различного типа и направления деятельности [13].

MS SQL Server – реляционная СУБД, функции:

- служба хранения, обработки и обеспечения целостности данных;

- служба для очистки данных, удаления дубликатов;
- модуль бизнес-аналитики;
- модуль построения отчетов и контроля правильности данных, на основе которых строится отчет;
- механизм репликации данных (синхронизация баз данных);
- поддержка, обработка и журналирование транзакций;
- собственный клиент с визуальным интерфейсом для взаимодействия с базой данных.

Решение от компании Microsoft распространяется как по коммерческой, так и свободной лицензии (Express-версия). СУБД рассчитана на системы любого типа и размера.

СУБД MySQL – свободно распространяемое программное обеспечение, рассчитанное на веб-приложения, а также небольшие и средние системы. Принадлежит компании Oracle. Существует коммерческая версия СУБД, которая может быть модернизирована под нужды определенного клиента. MySQL обладает наибольшей среди современных СУБД скоростью по обработке данных. Поддерживает несколько типов таблиц, список поддерживаемых типов постоянно расширяется. Каждый из типов обладает определенными особенностями. В рамках одной базы данных могут быть использованы разные тип таблиц, при необходимости. Имеет визуальный клиент, позволяющий проектировать и создавать базы данных на основе созданной при проектировании модели [9] – [10].

Результаты сравнения СУБД приведены в таблице 9.

Таблица 9 – Сравнение СУБД

	Свободно распространяемая версия	Ограничение на использование CPU	Защита данных	Скорость работы
Oracle	-/+	-	+	-
MS SQL Server	+	-	+	+
MySQL	+	-	+/-	+

Таким образом, для создания базы данных выбрана СУБД MS SQL Server, так как она обеспечивает надежную защиту хранимых данных, обладает высокой скоростью работы. Express-версия сервера обладает достаточно широкими функциональными возможностями для обработки и визуализации данных.

3.4 Разработка физической модели данных информационной системы по учету кредитных операций

Анализ предметной области позволил выявить следующие основные сущности:

- Клиент – клиент микрофинансовой организации;
- Страховка – оформляется к заявке по желанию клиента;
- Заявка – заявка на получение кредита;
- Документ – документ, прилагаемый к заявке, скан паспорта, претензия к должнику и т.д.;
- График оплаты – сроки выплат по кредиту и суммы, которые необходимо вносить;
- Менеджер – сотрудник, оформляющий заявку, либо работающий с должниками.

Физическая модель базы данных приведена на рисунке 13.

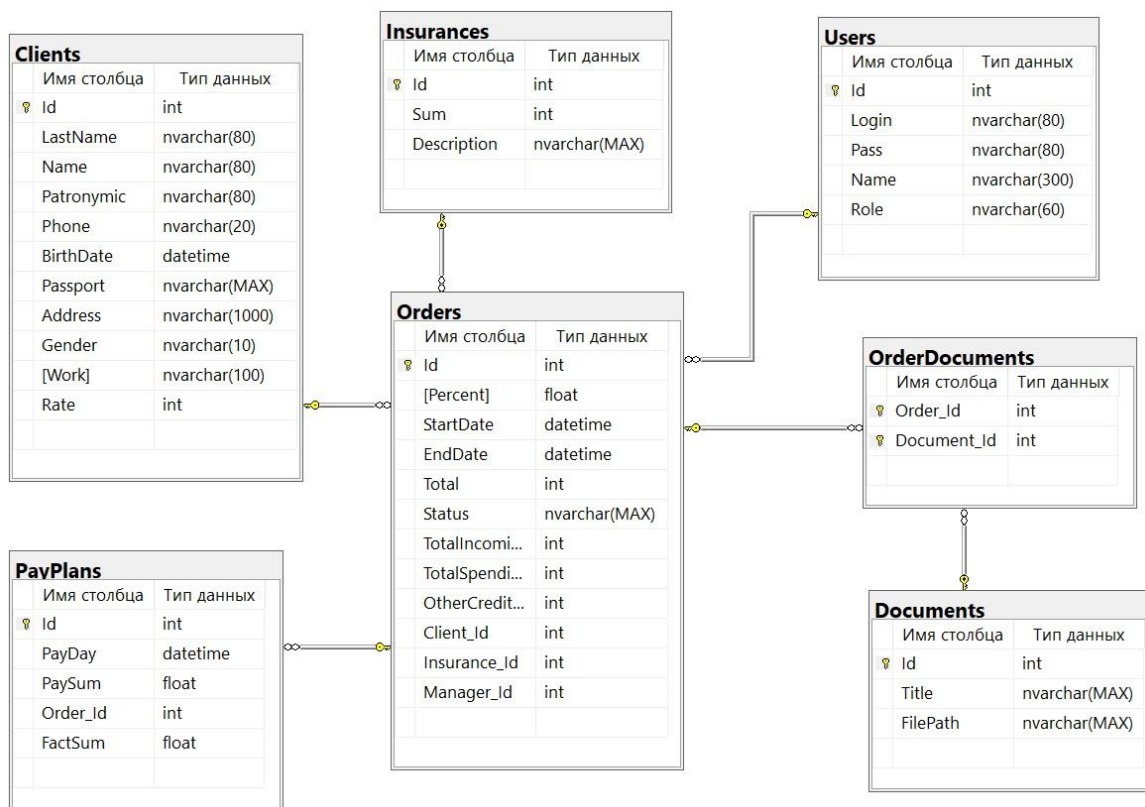


Рисунок 13 – Физическая модель базы данных

Характеристики атрибутов приведены в таблицах 10-16.

Таблица 10 – Users (Пользователи)

Атрибут	Тип данных	Описание
Id	int	Идентификатор пользователя, первичный ключ
Login	nvarchar(80)	Имя учетной записи
Pass	nvarchar(80)	Пароль от учетной записи
Name	nvarchar(300)	Имя пользователя
Role	nvarchar(60)	Тип учетной записи пользователя

Таблица 11 – Insurances (Страховки)

Атрибут	Тип данных	Описание
Id	int	Идентификатор страховки, первичный ключ
Sum	int	Сумма страховки
Description	nvarchar(MAX)	Описание условий получения страховки

Таблица 12 – Clients (Клиенты)

Атрибут	Тип данных	Описание
Id	int	Идентификатор клиента, первичный ключ
LastName	nvarchar(80)	Полное имя клиента
Name	nvarchar(80)	
Patronymic	nvarchar(80)	
Phone	nvarchar(20)	Номер телефона клиента
BirthDate	datetime	Дата рождения клиента
Passport	nvarchar(MAX)	Номер и серия паспорта клиента
Address	nvarchar(1000)	Адрес регистрации клиента
Gender	nvarchar(10)	Пол клиента
[Work]	nvarchar(100)	Должность клиента
Rate	int	Рейтинг

Таблица 13 – PayPlans (Графики платежей)

Атрибут	Тип данных	Описание
Id	int	Идентификатор записи, первичный ключ
PayDay	datetime	Дата платежа
PaySum	float	Сумма по плану
Order_Id	int	Номер заявки, внешний ключ
FactSum	float	Фактически внесенная клиентом сумма

Связь между таблицами Orders и PayPlans относится к типу «один-ко-многим»: каждая запись в графике соответствует только одной заявке, одной заявке могут соответствовать несколько записей в графике.

Таблица 14 – Documents (Документы)

Атрибут	Тип данных	Описание
Id	int	Идентификатор документа, первичный ключ
Title	nvarchar(MAX)	Название документа
FilePath	nvarchar(MAX)	Путь к файлу с документом

Таблица 15 – OrderDocuments (Документы заявки)

Атрибут	Тип данных	Описание
Order_Id	int	Номер заявки, внешний ключ, часть составного первичного ключа
Document_Id	int	Номер документа, внешний ключ, часть составного первичного ключа

Таблица OrderDocuments используется для разделения связи типа «многие-ко-многим» между таблицами Orders и Documents, так как одной заявке могут соответствовать несколько документов, а документы одного типа могут соответствовать нескольким заявкам.

Таблица 16 – Orders (Заявки)

Атрибут	Тип данных	Описание
Id	int	Идентификатор заявки, первичный ключ
[Percent]	float	Процент выдачи кредита
StartDate	datetime	Дата выдачи кредита
EndDate	datetime	Дата погашения кредита
Total	int	Запрашиваемая клиентом сумма
Status	nvarchar(MAX)	Статус заявки
TotalIncoming	int	Общий доход клиента за месяц
TotalSpending	int	Общие траты клиента за месяц
OtherCreditsAmount	int	Количество других кредитов
Client_Id	int	Номер клиента, внешний ключ
Insurance_Id	int	Номер страховки, внешний ключ
Manager_Id	int	Номер менеджера, оформившего заявку, внешний ключ

Связь между таблицами Managers и Orders относится к типу «один-ко-многим»: менеджер может оформить несколько заявок, каждая заявка может быть оформлена только одним менеджером.

Связь между таблицами Insurances и Orders относится к типу «один-ко-многим»: страховка одного типа может относиться к нескольким заявкам, к каждой заявке может относиться страховка только одного типа.

MS SQL Server использует диалект языка SQL, называемый Transact-SQL или T-SQL. Данный диалект создан компанией Microsoft и является расширением стандартного SQL. T-SQL позволяет:

- объявлять локальные и глобальные переменные;
- использовать управляющие операторы;
- использовать средства аутентификации Microsoft Windows
- применять множество дополнительных функций для работы с датами, строками, числовыми значениями.

Одной из особенностей T-SQL является применение директив сценария, предназначенных для управления выполнением указанного сценария. Например, директива GO указывает на окончание пакета инструкций и сигнализирует о начале выполнения данных инструкций. Директива EXEC указывает на вызов хранимой процедуры и т.д.

3.5 Разработка программного обеспечения информационной системы по учету кредитных операций

3.5.1 Схема взаимосвязи модулей приложения информационной системы по учету кредитных операций

Все функции информационной системы разделены на следующие группы:

- служебные: функции, необходимые для корректной работы информационной системы, но не имеющие прямого отношения к предметной области.
- основные – функции, направленные на обработку исходных данных и их представление пользователю на формах и отчетах.

Основные функции, в свою очередь, подразделяются на:

- функции сбора данных: получение информации из базы;
- функции обработки данных: формирование и обработка заявок, построение отчетов и т.д.;

– функции представления данных: отображение обработанной информации в отчетах и на экранных формах.

На рисунке 14 показано дерево функций информационной системы.



Рисунок 14 – Дерево функций ИС

Сценарий диалога для информационной системы для микрофинансовой организации приведен на рисунке А.1 в приложении А.

После успешной авторизации пользователь, в зависимости от типа учетной записи, будет перенаправлен в соответствующий личный кабинет. Далее с помощью меню пользователь может запускать на выполнение функции информационной системы.

3.5.2 Описание модулей приложения АИС с примерами программного кода

Дерево программных модулей показано на рисунке Б.1 (приложение Б).

Как видно из рисунка, стартовым модулем является App.xaml, в котором указан стартовый модуль при запуске приложения. Стартовым модулем является форма авторизации, далее следуют модули личных кабинетов и подчиненные им модули.

3.6 Описание функциональности информационной системы по учету кредитных операций

При запуске приложения пользователь видит форму авторизации (рисунок 15).

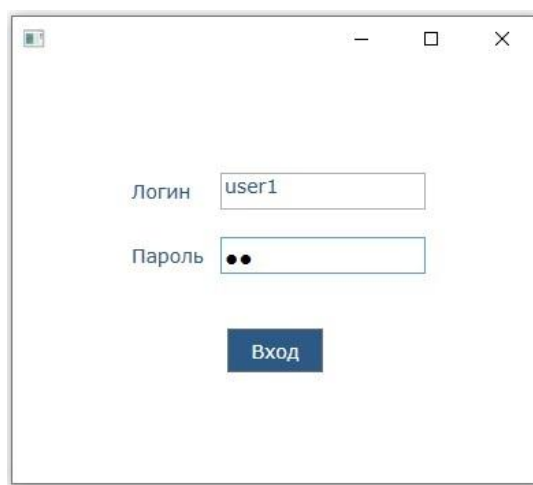
The image shows a screenshot of a web application's login form. It is contained within a window with standard OS window controls (minimize, maximize, close) in the top right corner. The form has a white background and contains three main elements: a label 'Логин' (Login) followed by a text input field containing the text 'user1'; a label 'Пароль' (Password) followed by a password input field with two black dots representing masked characters; and a blue rectangular button with the white text 'Вход' (Login) centered below the password field.

Рисунок 15 – Форма авторизации

При введении неверного логина или пароля система отобразит сообщение об ошибке (рисунок 16).

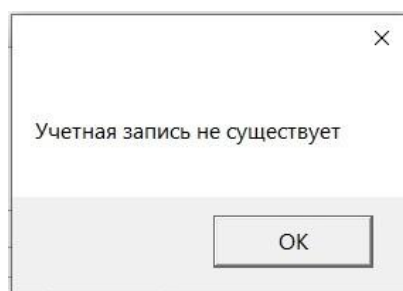


Рисунок 16 – Ошибка авторизации

Сценарий диалога, демонстрирующий На рисунке 17 показан личный кабинет менеджера по работе с клиентами, вкладка «Клиенты».

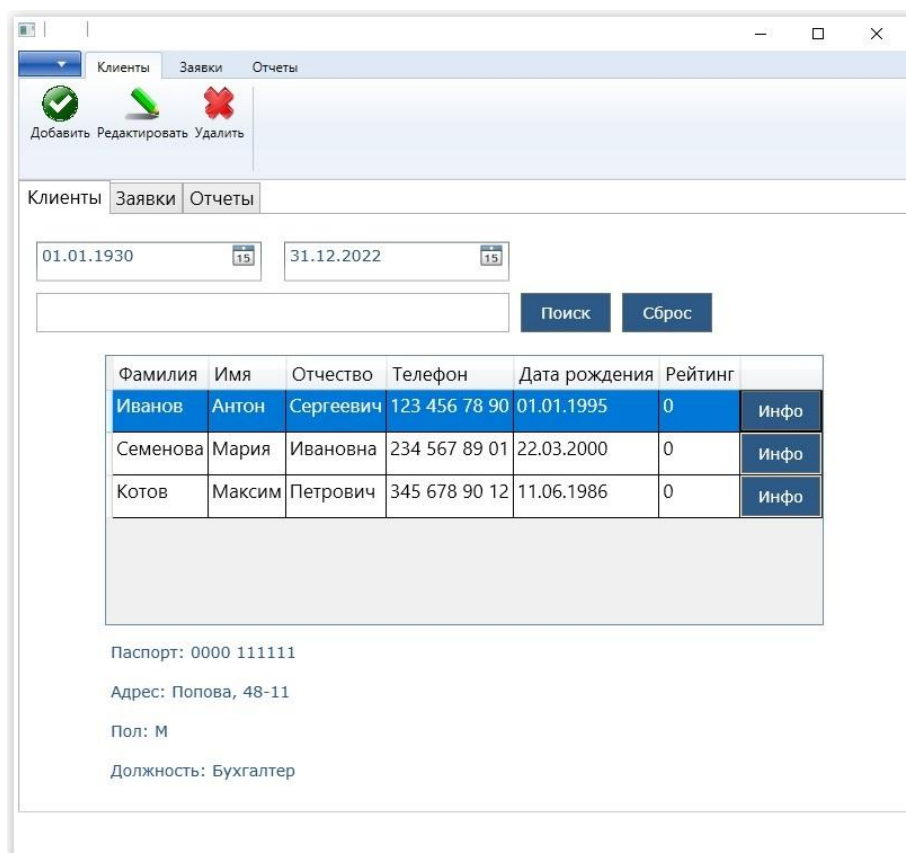


Рисунок 17 – Личный кабинет менеджера по работе с клиентами, вкладка «Клиенты»

При нажатии на кнопку «Инфо» в таблице с данными в нижней части окна отображается дополнительная информация по выбранному клиенту. В верхней части окна расположена форма поиска. Пример работы функции поиска показан на рисунке 18.

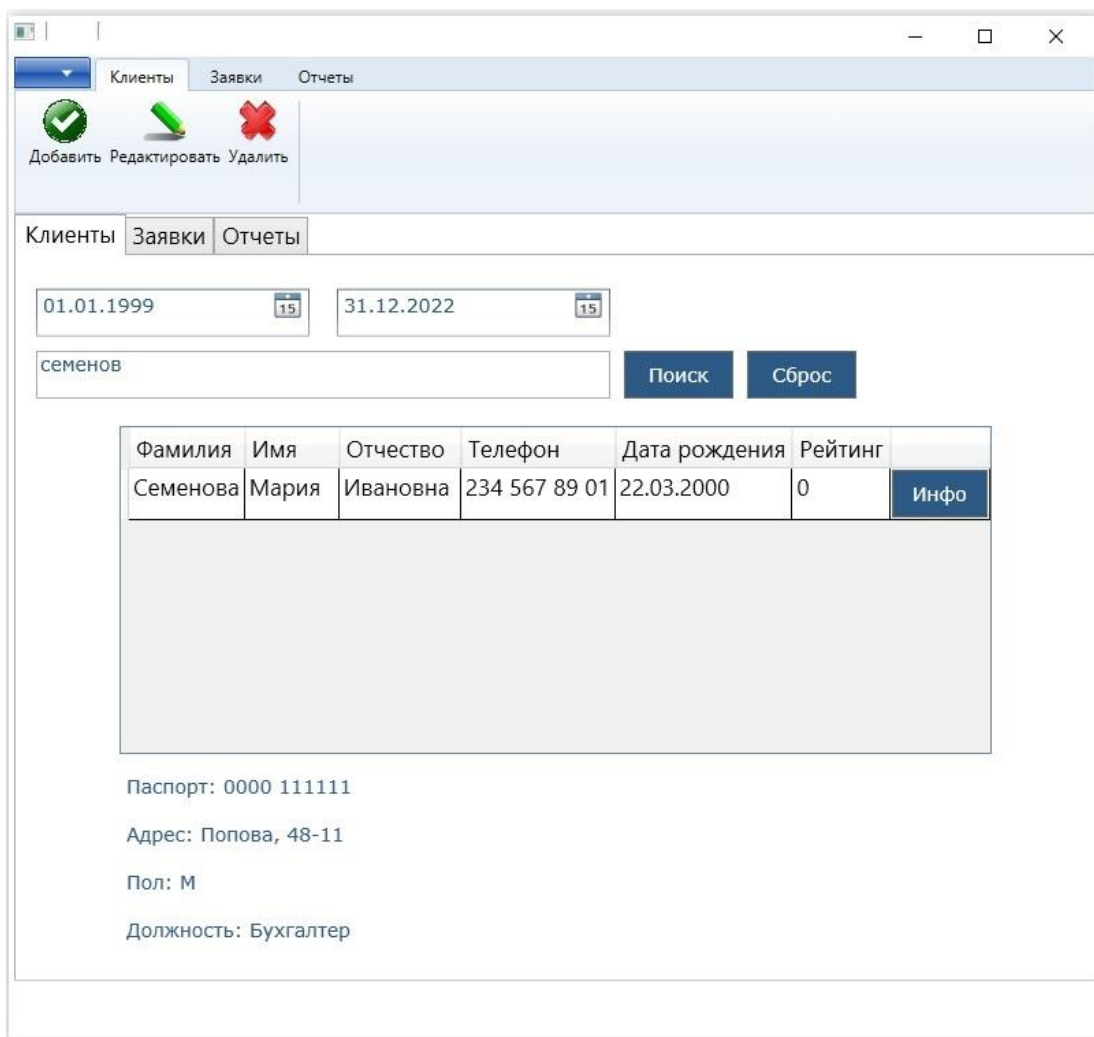


Рисунок 18 – Поиск по дате рождения и фамилии

Поиск выполняется по следующим параметрам: дата рождения, фамилия, номер паспорта, телефон, адрес. На рисунке 19 показано одержимое выпадающего меню в верхней части окна.



Рисунок 194 – Выпадающее меню

Под выпадающим меню расположено основное меню с командами для добавления, редактирования и удаления записей в таблице. При выборе команды «Добавить» пункта меню «Клиенты» откроется форма для добавления нового клиента (рисунок 20).

A screenshot of a software application window titled 'Новый клиент' (New Client). The window contains several input fields organized into sections: 'ФИО' (Full Name) with sub-fields for 'Фамилия' (Surname), 'Имя' (Name), and 'Отчество' (Patronymic); 'Контакты' (Contacts) with fields for 'Телефон' (Phone) and 'Адрес' (Address); 'Паспортные данные' (Passport Data) with fields for 'Дата рождения' (Date of Birth) containing '15.04.2022', 'Паспорт' (Passport), and 'Пол' (Gender) with a dropdown arrow; and 'Работа' (Work) with a field for 'Должность' (Position). A blue 'Добавить' (Add) button is located at the bottom center of the form.

Рисунок 20 – Форма добавления данных

Для изменения выбранной в таблице записи необходимо кликнуть по пункту «Редактировать». Если запись не выбрана, то пользователь увидит сообщение об ошибке (рисунок 21).

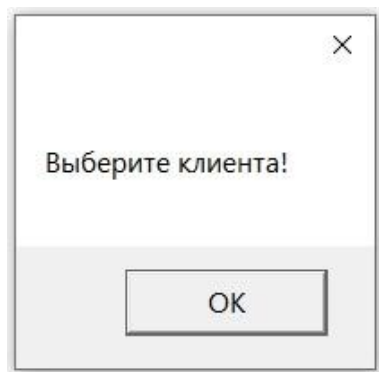


Рисунок 21 – Сообщение об ошибке редактирования

Форма редактирования выбранной записи показана на рисунке 22.

A screenshot of a web form titled "Новый клиент" (New client). The form is organized into four main sections: "ФИО" (Full Name) with fields for "Фамилия" (Last name: Иванов), "Имя" (First name: Антон), and "Отчество" (Patronymic: Сергеевич); "Контакты" (Contacts) with fields for "Телефон" (Phone: 123 456 78 90) and "Адрес" (Address: Попова, 48-11); "Паспортные данные" (Passport data) with fields for "Дата рождения" (Date of birth: 01.01.1995), "Паспорт" (Passport: 0000 111111), and "Пол" (Gender: M); and "Работа" (Work) with a field for "Должность" (Position: Бухгалтер). At the bottom of the form is a blue button labeled "Добавить" (Add).

Рисунок 22 – Форма редактирования записи о клиенте

Вкладка «Заявки» содержит форму поиска по различным параметрам, таблицу с заявками, список дополнительных параметров. Поиск работает по следующим параметрам:

- полное имя и идентификатор клиента;
- период создания заявки;
- наличие задолженности.

Вкладка «Заявки» и соответствующие пункты меню показаны на рисунке 23.

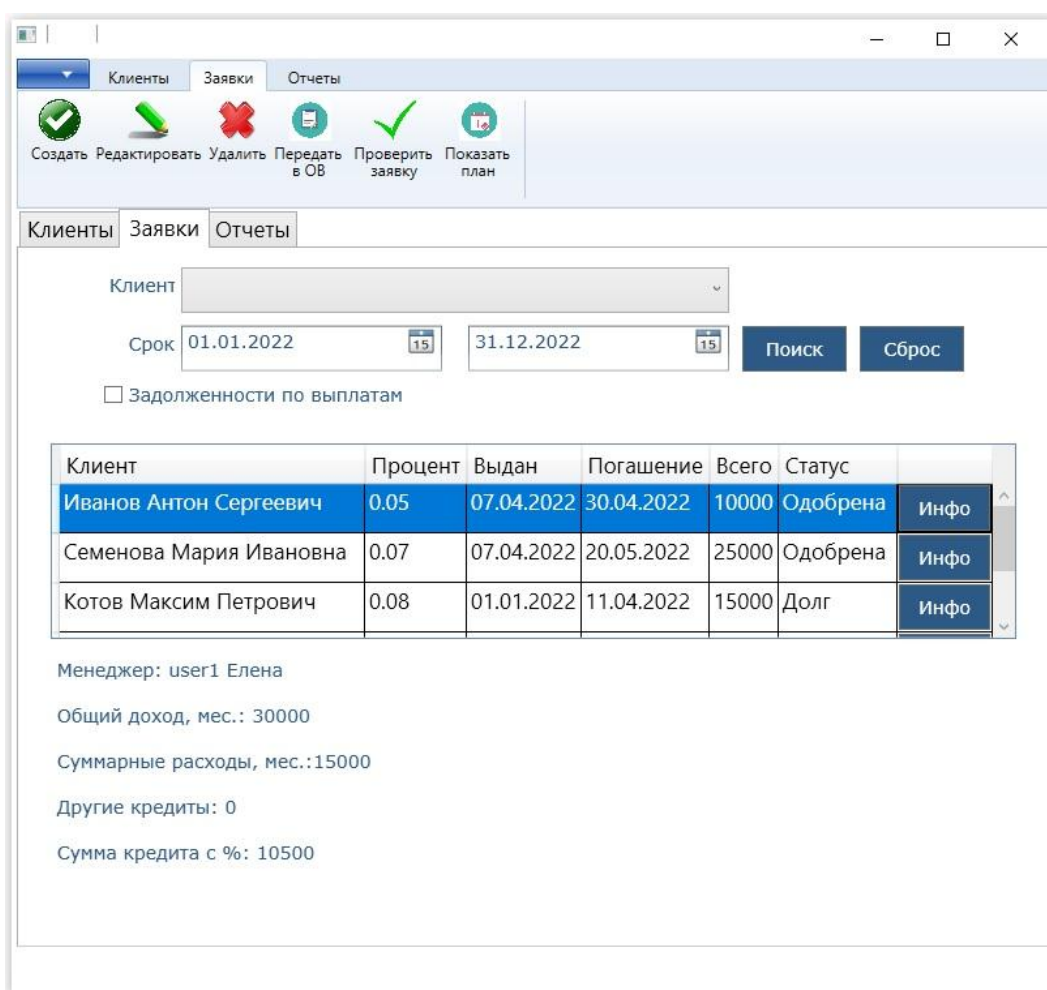


Рисунок 23 – Личный кабинет менеджера по работе с клиентами, вкладка «Заявки»

Дополнительные параметры появляются при нажатии на кнопку «Инфо» в таблице. Кнопка «Сброс» предназначена для очистки полей поиска. Поиск заявок с задолженностями выполняется с помощью чекбокса «Задолженности по выплатам» (рисунок 24).

Клиент	Процент	Выдан	Погашение	Всего	Статус	
Котов Максим Петрович	0.08	01.01.2022	11.04.2022	15000	Долг	Инфо

Рисунок 24 – Поиск заявок с долгами

Для работы с заявками используются следующие пункты меню: «Создать», «Редактировать», «Удалить», «Передать в ОВ», «Проверить заявку», «Показать план».

Форма для добавления новой заявки показана на рисунке 25.

The screenshot shows a web form titled "Новая заявка" (New Application) with the following fields:

- Клиент**: A dropdown menu.
- Срок выдачи**: Two date pickers, both set to 15.04.2022.
- Сумма**: Two input fields labeled "Процент" and "Желаемая сумма".
- Доходы и расходы**: Two input fields labeled "Суммарный доход" and "Суммарный расход".
- Кредиты и страховка**: Three input fields: "Другие кредиты", "Сумма страховки", and "Описание страховки" (a dropdown menu).

A blue "Добавить" (Add) button is located at the bottom center of the form.

Рисунок 25 – Форма добавления новой заявки

Форма редактирования записи показана на рисунке 26.

The screenshot shows the same "Новая заявка" form, but with data entered into the fields:

- Клиент**: "Иванов Антон Сергеевич" (selected in the dropdown).
- Срок выдачи**: Date pickers set to 07.04.2022 and 30.04.2022.
- Сумма**: "Процент" is 0,05; "Желаемая сумма" is 10000.
- Доходы и расходы**: "Суммарный доход" is 30000; "Суммарный расход" is 15000.
- Кредиты и страховка**: "Другие кредиты" is 0; "Сумма страховки" and "Описание страховки" are empty.

The blue "Добавить" (Add) button remains at the bottom center.

Рисунок 26 – Форма для редактирования новой записи

Пункт меню «Удалить» предназначен для удаления заявки. Пункт «Проверить заявку» позволяет оценить надежность заемщика и одобрить заявку или отклонить ее. Результат проверки отображается в виде сообщения (рисунок 27), а также указывается в статусе заявки.

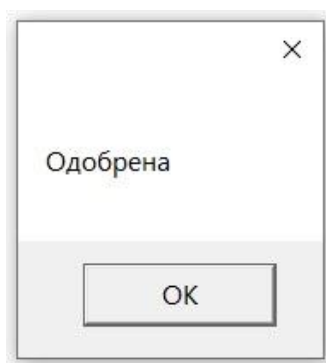
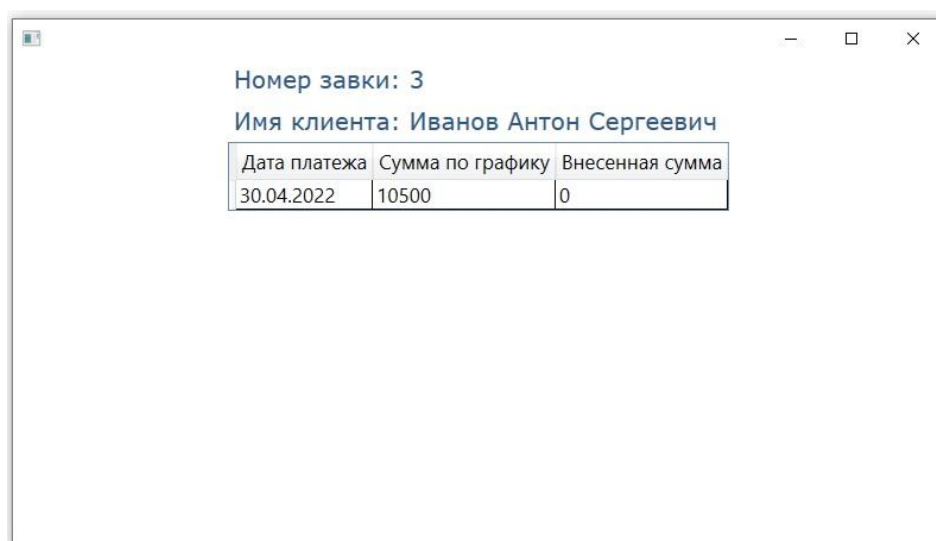


Рисунок 27 – Сообщение о результате проверки заявки

Пункт меню «Передать в ОВ» предназначен для изменения статуса заявки на «Долг» и передачи в отдел взыскания. Пункт «Показать план» позволяет отобразить план по выплатам по выбранной заявке (рисунок 28).



Номер заявки: 3		
Имя клиента: Иванов Антон Сергеевич		
Дата платежа	Сумма по графику	Внесенная сумма
30.04.2022	10500	0

Рисунок 28– План выплат по заявке

Вкладка «Отчеты» позволяет выбрать тип отчета и временной период для построения отчета (рисунок 29).

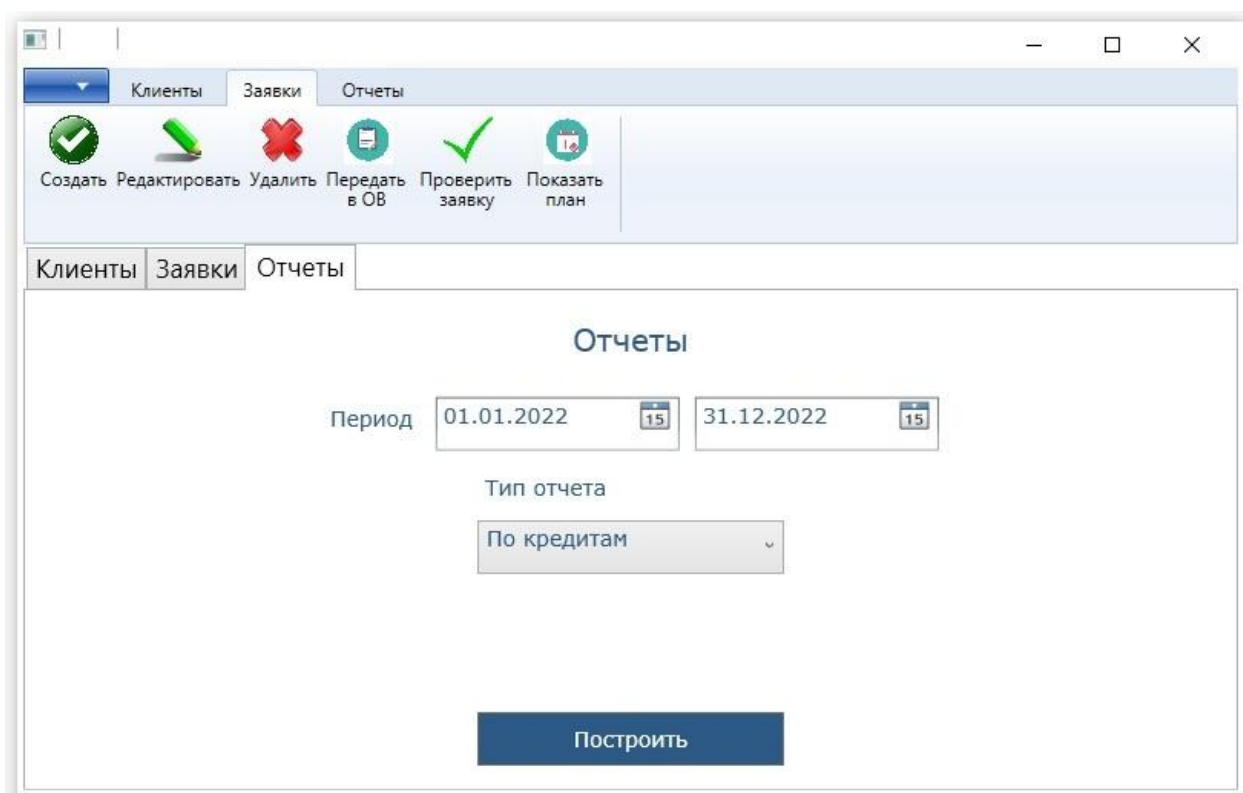


Рисунок 29 – Вкладка «Отчеты», выбран тип «По кредитам»

При выборе типа отчета «По клиентам» на форме появляется дополнительный список с клиентами (рисунок 30).

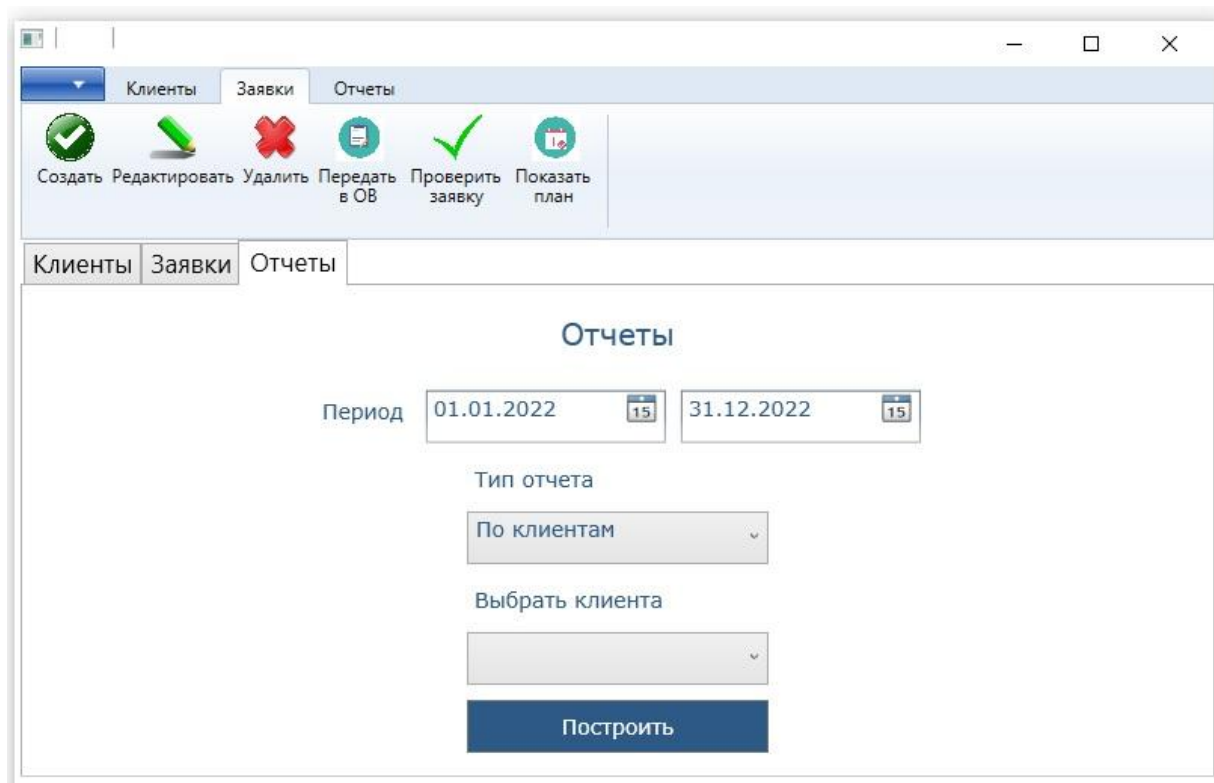


Рисунок 30 – Вкладка «Отчеты», выбран тип «По клиентам»

На рисунке 31 показан личный кабинет менеджера по взысканиям, вкладка «Долги».

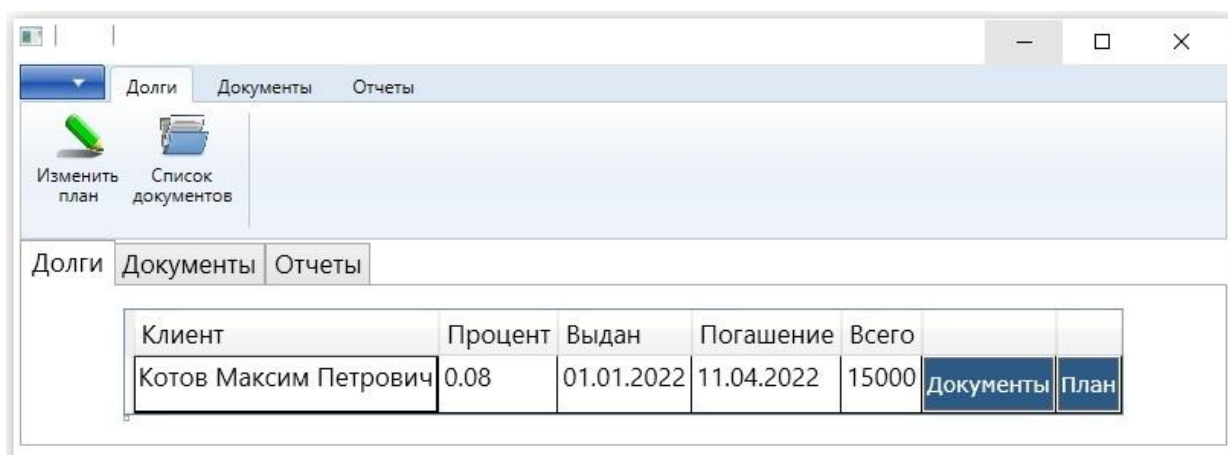
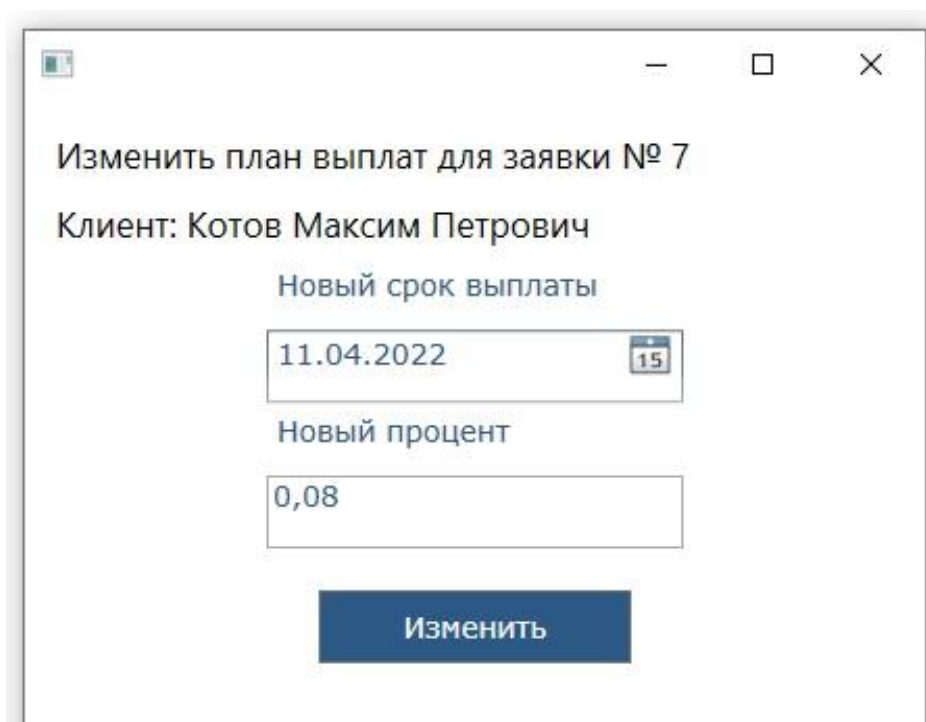


Рисунок 51 – Личный кабинет менеджера по взысканиям, вкладка «Долги»

При нажатии на пункт меню «Список документов» под таблицей с заявками появятся документы, соответствующие заявке. Пункт меню «Изменить план» позволяет перестроить план выплат. План перестраивается при увеличении срока возврата или изменении процента. Форма изменения плана показана на рисунке 32.



Изменить план выплат для заявки № 7

Клиент: Котов Максим Петрович

Новый срок выплаты

11.04.2022

Новый процент

0,08

Изменить

Рисунок 32 – Форма изменения плана

На рисунке 33 показана вкладка «Документы».

Долги | Документы | Отчеты

Создать

Долги | Документы | Отчеты

Документы

Тип документа

Клиент

Заявка

Построить

Рисунок 33 – Вкладка «Документы»

Данная вкладка позволяет создать документ. Для создания необходимо выбрать тип документа, клиента и заявку. На рисунке 34 показана вкладка «Отчеты».

Долги | Документы | Отчеты

Построить

Долги | Документы | Отчеты

Отчеты

Период 01.01.2022 31.12.2022

Тип отчета

Построить

Рисунок 34 – Вкладка «Отчеты»

Для создания отчета необходимо выбрать его тип и временной период. Затем нажать кнопку «Построить». Примеры отчетов показаны на рисунках В.1 - В.4 в приложении В.

3.7 Оценка и обоснование экономической эффективности разработки информационной системы

3.7.1 Выбор методики расчета экономической эффективности

Для обоснования экономической эффективности разработанной информационной системы использована методика расчета прямой эффективности от внедрения информационной системы. Методика заключается в сравнении базового и предлагаемого варианта обработки информации.

В качестве базового варианта выступает ручная обработка данных с использованием табличных процессоров и текстовых редакторов. В качестве предлагаемого варианта используется обработка данных с использованием информационной системы. Основой расчетов являются временные затраты рабочего времени сотрудников, стоимость рабочего времени. Стоимость одного часа рабочего времени сотрудника составляет 291 р. из расчета заработной платы с учетом отчислений в 46550 р.

3.7.2 Расчет показателей экономической эффективности проекта

К автоматизируемым операциям относятся: формирование заявки, проверка заявки, оформление отчетов и документов. Временные затраты на выполнение указанных операций при отсутствии и наличии автоматизации приведены в таблицах.

Стоимость затрат на обработку информации определяется на основе затрат на заработную плату сотрудника организации, выполняющего обработку данных. Величины затрат на обработку информации по базовому и предлагаемому вариантам приведены в таблицах 17-20.

Таблица 17 – Показатели эффективности от внедрения проекта для формирования заявки

	Затраты		Абсолютное изменение затрат	Коэффициент изменения затрат	Индекс изменения затрат
	Базовый вариант	Проектный вариант			
	T_0 (мин)	T_1 (мин)			
Трудоемкость	T_0 (мин)	T_1 (мин)	$\Delta T = T_0 - T_1$	$K_T = \frac{\Delta T}{T_0} \cdot 100\%$	$Y_T = \frac{T_0}{T_1}$
	7	5	2	28,6	1,4
Стоимость	C_0 (руб)	C_1 (руб)	$\Delta C = C_0 - C_1$ (руб)	$K_C = \frac{\Delta C}{C_0} \cdot 100\%$	$Y_C = \frac{C_0}{C_1}$
	4,2	3,0	1,2	28,6	1,4

Таблица 18 – Показатели эффективности от внедрения проекта для проверки заявки

	Затраты		Абсолютное изменение затрат	Коэффициент изменения затрат	Индекс изменения затрат
	Базовый вариант	Проектный вариант			
	T_0 (мин)	T_1 (мин)			
Трудоемкость	T_0 (мин)	T_1 (мин)	$\Delta T = T_0 - T_1$	$K_T = \frac{\Delta T}{T_0} \cdot 100\%$	$Y_T = \frac{T_0}{T_1}$
	15	5	10	66,7	3
Стоимость	C_0 (руб)	C_1 (руб)	$\Delta C = C_0 - C_1$ (руб)	$K_C = \frac{\Delta C}{C_0} \cdot 100\%$	$Y_C = \frac{C_0}{C_1}$
	9,1	3	6,1	66,7	3

Таблица 19 – Показатели эффективности от внедрения проекта для построения отчета

	Затраты		Абсолютное изменение затрат	Коэффициент изменения затрат	Индекс изменения затрат
	Базовый вариант	Проектный вариант			
	T_0 (мин)	T_1 (мин)			
Трудоемкость	T_0 (мин)	T_1 (мин)	$\Delta T = T_0 - T_1$	$K_T = \frac{\Delta T}{T_0} \cdot 100\%$	$Y_T = \frac{T_0}{T_1}$
	120	4	116	96,7	30
Стоимость	C_0 (руб)	C_1 (руб)	$\Delta C = C_0 - C_1$ (руб)	$K_C = \frac{\Delta C}{C_0} \cdot 100\%$	$Y_C = \frac{C_0}{C_1}$
	72,8	2,4	70,3	96,7	30,0

Таблица 20 – Показатели эффективности от внедрения проекта для оформления документа

	Затраты		Абсолютное изменение затрат	Коэффициент изменения затрат	Индекс изменения затрат
	Базовый вариант	Проектный вариант			
Трудоемкость	T_0 (мин)	T_1 (мин)	$\Delta T = T_0 - T_1$	$K_T = \frac{\Delta T}{T_0} \cdot 100\%$	$Y_T = \frac{T_0}{T_1}$
	20	3	17	85	6,7
Стоимость	C_0 (руб)	C_1 (руб)	$\Delta C = C_0 - C_1$ (руб)	$K_C = \frac{\Delta C}{C_0} \cdot 100\%$	$Y_C = \frac{C_0}{C_1}$
	12,1	1,8	10,3	85,0	6,7

Как видно из таблиц, использование разработанной информационной системы позволяет снизить временные затраты на обработку информации. Снижение затрат по времени позволит сотрудникам решать большее количество задач за рабочий день и, таким образом, увеличить прибыль организации.

Выводы по главе 3

В третьей главе приведено описание процесса разработки информационной системы для микрофинансовой организации. Обоснован выбор архитектуры системы, технологий разработки. Представлена физическая модель базы данных, а также материалы, на основе которых данная модель построена. Приведены скриншоты интерфейса разработанной информационной системы. Представлены результаты расчета экономической эффективности от внедрения системы, показывающие, что внедрение информационной системы позволит сократить время на решение рутинных задач.

Заключение

Разработка информационной системы для учета кредитных операций микрофинансовой организации является актуальной задачей.

В ходе выполнения выпускной квалификационной работы решены следующие задачи:

- проведен анализ предметной области – процесса учета кредитных операций в микрофинансовой организации;
- построена модель бизнес-процесса по учету кредитных операций в микрофинансовой организации;
- проведен обзор готовых решений;
- обоснована необходимость разработки собственной системы;
- сформулирована задача на разработку информационной системы;
- спроектирована база данных информационной системы учета кредитных операций;
- спроектирован интерфейс к базе данных;
- разработана информационная система по учету кредитных операций в микрофинансовой организации;
- оценена экономическая эффективность от внедрения разработанной системы.

Для разработки использованы следующие инструменты:

- объектно-ориентированный язык программирования C#;
- среда разработки Visual Studio 2019 Community;
- реляционная СУБД MS Sql Server Express;
- фреймворк для работы с базами данных Entity Framework;
- структурные паттерны проектирования MVVM, Singleton.

Таким образом можно сделать вывод о том, что поставленные задачи решены, цель достигнута.

Список используемой литературы и используемых источников

1. Абрамян, М. Visual C# на примерах / М. Абрамян. – М.: БХВ-Петербург, 2016. – 436 с.
2. Бадд, Т. Объектно-ориентированное программирование в действии / Т. Бадд М.: Питер. 2016. – 400 с.
3. Бачурина, С.С. Информационное моделирование / С.С. Бачурина. – М.: ДМК, 2021. – 112 с.
4. Вагнер, Б. C# Эффективное программирование / Б. Вагнер. – М.: ЛОРИ, 2017. – 320 с.
5. Вайсфельд, М. Объектно-ориентированный подход / М. Вайсфельд – М.: Питер, 2020. – 256 с.
6. Воронцова, Л В., Евсеев Г.В. Микрофинансирование как инструмент финансово-кредитной системы: опыт, проблемы и перспективы развития в России // Russian Journal of Economics and Law. 2012. №1 (21). URL: <https://cyberleninka.ru/article/n/mikrofinansirovanie-kak-instrument-finansovo-kreditnoy-sistemy-opyt-problemy-i-perspektivy-razvitiya-v-rossii> (дата обращения: 16.04.2022).
7. Громов, А.И. Управление бизнес-процессами: современные методы. монография / А.И. Громов, А. Фляйшман, В. Шмидт. – Люберцы: Юрайт, 2016. – 367 с.
8. Гросс, Кр. C# 2008 и платформа .NET 3.5 Framework / Кр. Гросс. – М.: Вильямс, 2016. – 480 с.
9. Грофф, Дж., SQL. Полное руководство/ Дж. Грофф, П. Вайнберг – М.: Вильямс, 2018. – 960 с.
10. Дейт, К. Введение в системы баз данных / К. Дейт – М.: Юрайт, 2019. – 784 с.
11. Долганова, О.И. Моделирование бизнес-процессов: Учебник и практикум для академического бакалавриата / О.И. Долганова, Е.В. Виноградова, А.М. Лобанова. – Люберцы: Юрайт, 2016. – 289 с.

12. Ишкова, Э.А. Самоучитель C#. Начала программирования / Э.А. Ишкова. – М.: Наука и техника, 2017. – 496 с.
13. Конолли, Т., Базы данных. Проектирование, реализация и сопровождение. Теория и практика. / Т. Конолли, К. Бегг – М.: Вильямс, 2017. – 1440 с.
14. Культин, Н. C# в задачах и примерах / Н. Культин. – М.: БХВ-Петербург, 2016. – 952 с.
15. Макконел, С. Совершенный код / С. Макконел. – М.: Русская редакция, 2017. – 896 с.
16. Рихтер, Дж. CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C# / Дж. Рихтер. – М.: Питер, 2016. – 365 с.
17. Троелсен, Э. Язык программирования C# 5.0 и платформа .NET 4.5 / Э. Троелсен. – М.: Вильямс, 2019. – 486 с.
18. Шилов, В.А. Проблемы кредитования субъектов малого и среднего предпринимательства в России // АНИ: экономика и управление. 2017. №1 (18). URL: <https://cyberleninka.ru/article/n/problemy-kreditovaniya-subektov-malogo-i-srednego-predprinimatelstva-v-rossii> (дата обращения: 16.04.2022).
19. Albahari, B. C# in a Nutshell / B. Albahari, J. Albahari, – Washington: O'Reilly Media, 2017. – 1088p.
20. Albahari, J. Threading in C# / J. Albahari – Seb: O'Reilly Media, 2017. – 720 p.
21. Clark, D. Beginning C# Object-Oriented Programming / D. Clark – New York: Apress, 2018. – 780 p.
22. Liberty, J. Learning C# / J. Liberty – Seb: O'Reilly Media, 2018. – 630 p.
23. Richter, J. CLR via C# / J. Richter – Redmond: Microsoft Press, 2016. – 873 p.

Приложение А

Сценарий диалога для информационной системы

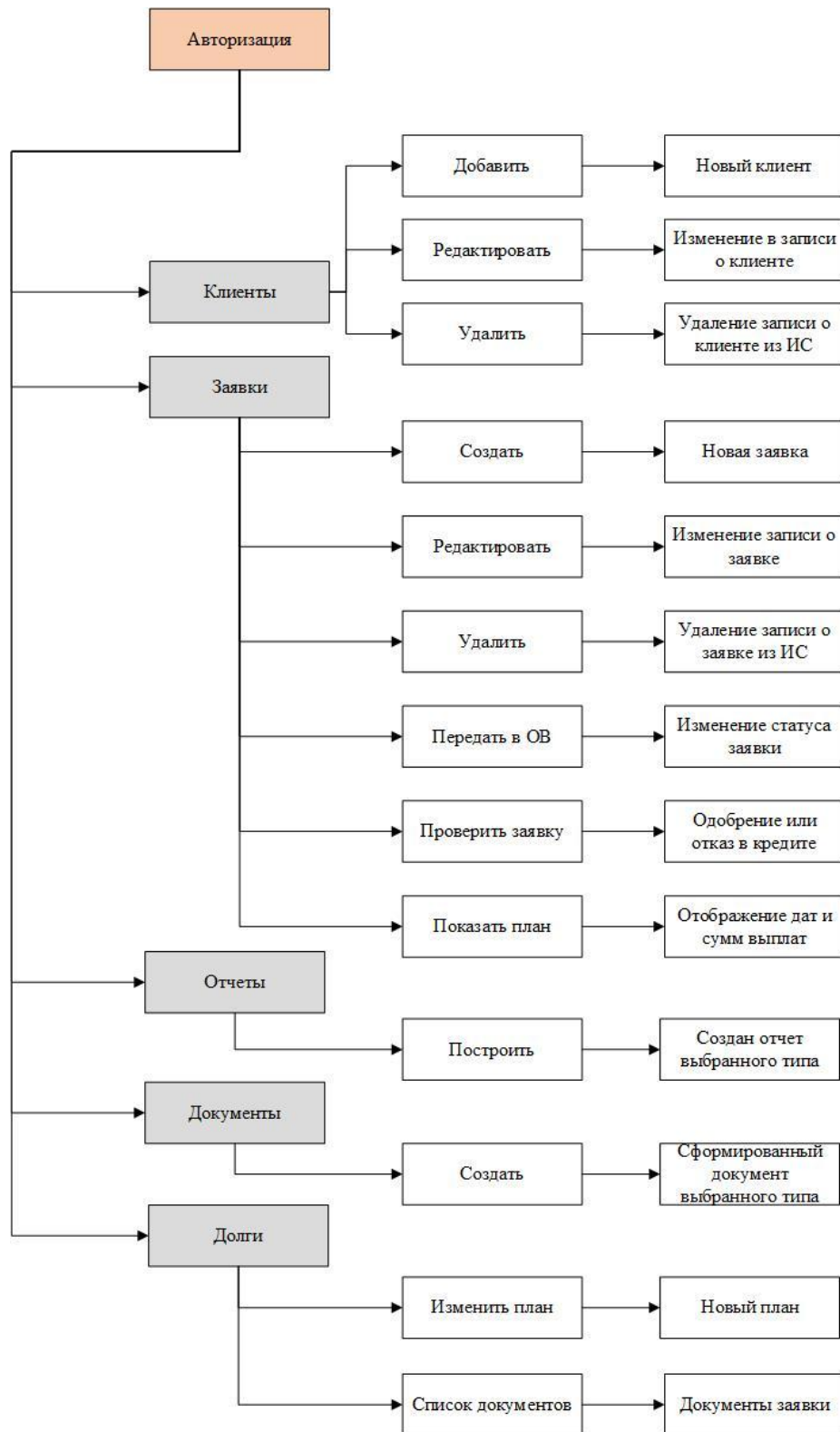


Рисунок А.1 – Схема пунктов меню для реализации диалога с пользователем

Приложение Б
Дерево программных модулей

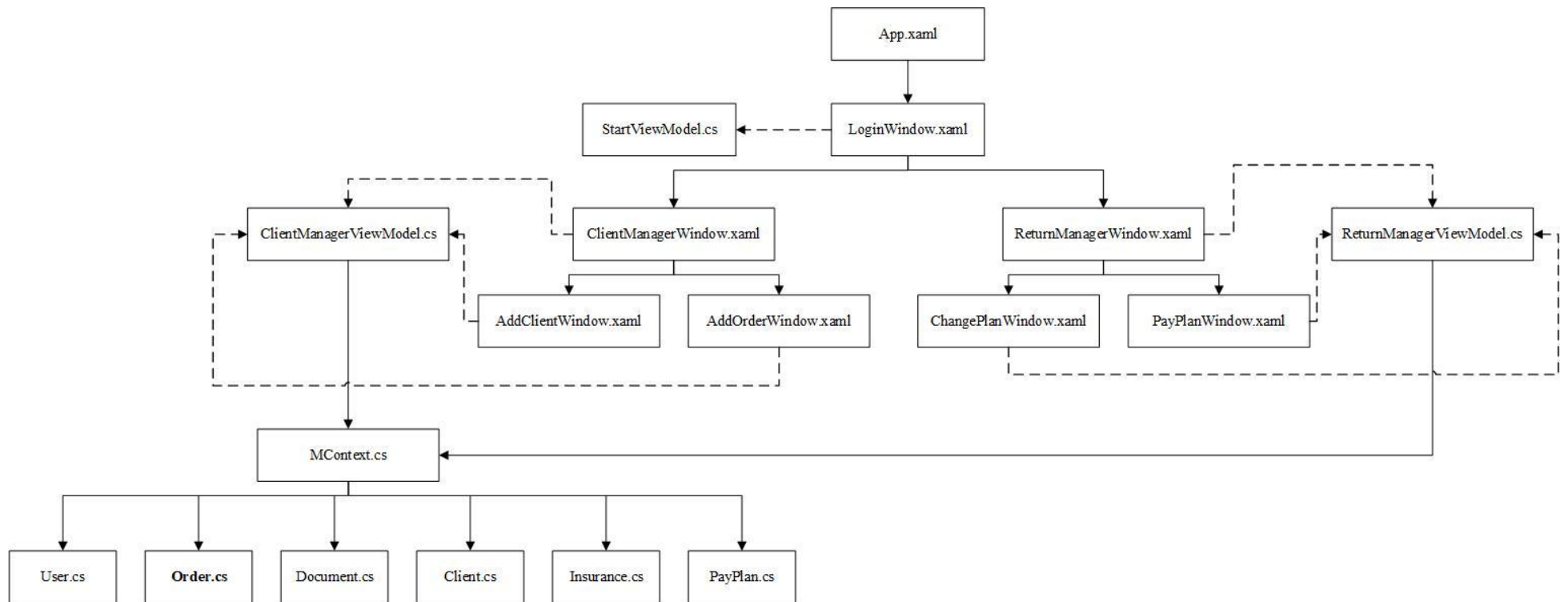


Рисунок Б.1 – Диаграмма компонентов для модулей приложения

Приложение В

Примеры отчетов

A	B	C	D	E	F	G
1			Отчет по займам			
2	Выбран период с 01.01.2022 по 31.12.2022					
3	Общая сумма	Количество клиентов	Количество своевременно выплаченных займов	Количество просроченных займов		
4	92300	5	0	1		
5						
6	Отчет сформирован: 16.04.2022 0:00:00					
7						

Рисунок В.6 – Отчет по кредитам, менеджер по работе с клиентами

A	B	C	D	E	F	G	H	I	J	K	L	M
1			Отчет по займам клиента									
2	Выбран период с 01.01.2022 по 31.12.2022											
3	ФИО	Всего	Просрочено	Выплачено	Открыто							
4	Иванов Антон Сергеевич	1	0	0	1							
5												
6	Отчет сформирован: 16.04.2022 0:00:00											
7												

Рисунок В.2 – Отчет по клиентам, менеджер по работе с клиентами

Продолжение Приложения В

A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Отчет по займам клиента												
2	Выбран период с 01.01.2022 по 31.12.2022												
3	ФИО	Открытые долги	Закрытые долги										
4	Клиент Семенова Мария Ивановна не имеет открытых или закрытых долгов												
5													
6	Отчет сформирован: 16.04.2022 0:00:00												
7													

Рисунок В.3 – Отчет по клиентам, менеджер по взысканиям

A	B	C	D	E	F	G	H	I	J
1	Отчет по займам								
2	Выбран период с 01.01.2022 по 31.12.2022								
3	Общая сумма займов	Количество открытых долгов	Количество закрытых долгов						
4	16200	1	0						
5									
6	Отчет сформирован: 16.04.2022 0:00:00								
7									

Рисунок В.7 – Отчет по кредитам, менеджер по взысканиям