Функциональная спецификация «Универсальный шаблон спецификации»

Для проекта команды номер 2 по решению группового кейса номер 4:  
«Международный банк “Глобал Финанс” ищет решение для автоматизации процесса риско-менеджмента и анализа кредитоспособности заемщиков.»

**Оглавление**

[1.](#_gjdgxs) Введение 4

[1.1.](#_30j0zll) Инструкции для авторов 4

[1.2.](#_1fob9te) История изменений 4

[1.3.](#_3znysh7) Блок согласований 4

[1.4.](#_2et92p0) Список рассылки 4

[2.](#_tyjcwt) Общее описание 4

[2.1.](#_3dy6vkm) Терминология 4

[2.2.](#_1t3h5sf) Соглашения, принятые в документах 4

[2.3.](#_4d34og8) Общий взгляд на продукт, бизнес-процесс 4

[2.4.](#_2s8eyo1) Обзор бизнес-требований 4

[2.5.](#_17dp8vu) Описание текущей системы 4

[2.6.](#_3rdcrjn) Предлагаемые методы решения 4

[2.7.](#_26in1rg) Взаимодействие пользовательских ролей 4

[2.8.](#_lnxbz9) Предположения, допущения, зависимости (проверить) 5

[2.9.](#_35nkun2) Границы проекта 5

[2.10.](#_1ksv4uv) Ссылки (хз надо заполнять или нет) 5

[3.](#_44sinio) Функциональность 5

[3.1.](#_2jxsxqh) Функциональные требования 5

[3.2.](#_z337ya) Варианты использования (Use Cases) 5

[3.3.](#_3j2qqm3) Пользовательские истории (User Stories) 5

[3.4.](#_1y810tw) Внутренние рабочие процессы системы 5

[3.5.](#_4i7ojhp) Перечень особенностей («фитч») или описания функций 5

[3.6.](#_2xcytpi) Функциональность администратора 5

[3.7.](#_1ci93xb) Обработка ошибок 5

[3.8.](#_3whwml4) Взаимодействие продукта (с другими продуктами и компонентами) 5

[4.](#_2bn6wsx) Требования к данным 5

[4.1.](#_qsh70q) Логическая модель данных 6

[4.2.](#_3as4poj) Словарь данных 6

[4.3.](#_1pxezwc) Отчеты 6

[4.4.](#_49x2ik5) Получение 6

[4.5.](#_2p2csry) Целостность 6

[4.6.](#_147n2zr) Хранение и утилизация данных; 6

[5.](#_3o7alnk) Требования к внешним интерфейсам 6

[5.1.](#_23ckvvd) Пользовательские интерфейсы 6

[5.1.1.](#_ihv636) Прототипы дизайнов 6

[5.1.2.](#_32hioqz) Контуры или наброски 6

[5.2.](#_1hmsyys) Интерфейсы ПО 6

[5.3.](#_41mghml) Интерфейсы оборудования 6

[5.4.](#_41mghml) Коммуникационные интерфейсы 6

[5.5.](#_2grqrue) Управление конфигурациями 6

[6.](#_vx1227) Платформа и фреймворки 6

[7.](#_3fwokq0) Тестирование и проверка 6

[8.](#_1v1yuxt) Критерий приёмки работ 7

[9.](#_4f1mdlm) Установка 7

[9.1.](#_2u6wntf) Настройка 7

[9.2.](#_19c6y18) Инсталляция 7

[10.](#_3tbugp1) Атрибуты 7

[10.1.](#_28h4qwu) Портативность 7

[10.2.](#_nmf14n) Производительность 7

[10.3.](#_37m2jsg) Безопасность 7

[10.4.](#_1mrcu09) Расширяемость 7

[10.5.](#_46r0co2) Интернационализация 7

[10.6.](#_2lwamvv) Поддержка и техническое обслуживание 7

[10.7.](#_111kx3o) Справка и документация 7

[10.8.](#_3l18frh) Надежность 7

[10.9.](#_206ipza) Доступность 7

[10.10.](#_4k668n3) Требования к удобству использования 7

# Введение

Эта спецификация требований к программному обеспечению предоставляет исчерпывающее описание как функциональных, так и нефункциональных требований, необходимых для успешного завершения выпускного проекта. Документ предназначен для использования командой разработчиков, которая будет осуществлять реализацию системы, а также специалистами по качеству, которые займутся проверкой её корректности и соответствия заявленным требованиям. За исключением случаев, специально оговоренных в тексте, все требования, изложенные в данном документе, обладают высоким приоритетом и должны быть выполнены в первую очередь.

## 1.1. Инструкции для авторов

Внесение любых значимых правок отражается в таблице пункта 1.2 История изменений.

После согласования документа 15.08.2022 внесение в документ изменений, которые оказывают существенное влияние на конечный продукт, преследуют исполнение не согласованных с заказчиком целей или иным образом уводят команду от реализации системы, ориентировочный образ которой согласован с заказчиком, считается нарушением договора подряда на создание ИС.

## 1.2. История изменений

|  |  |  |  |
| --- | --- | --- | --- |
| Версия | Описание правки | Дата | Автор |
| 1.0 | Описание FSD | 05.09.24 | all |
| 1.2 | Сбор функциональных и нефункциональных требований | 11.09.24 | Ростислав |
| 1.3 | Use case + User Stories | 15.09.24 | Эльза |
| 1.4.0 | Проектирование базы данных | 20.09.24 | Эльза, Алексей |
| 1.4.1 | Наполнение базы данных | 25.09.24 | all |
| 1.4.2 | Прототипирование | 26.09.24 | Алексей |
| 1.4.3 | Тестирование БД | 30.09.24 | Константин |
| 2.0 | Разработка презентации | 09.10.24 | all |

## 1.3. Блок согласований

|  |  |  |  |
| --- | --- | --- | --- |
| Версия | Дата | Статус | Автор |
| 1.0 | 07.09.24 | Утверждён | Марина |
| 1.2 | 12.09.24 | Утверждён | Марина |
| 1.3 | 17.09.24 | Утверждён | Марина |
| 1.4.0 | 23.09.24 | Утверждён | Марина |
| 1.4.1 | 27.09.24 | Утверждён | Марина |
| 1.4.2 | 28.09.24 | Утверждён | Марина |
| 1.4.3 | 01.10.24 | Утверждён | Марина |
| 2.0 | 10.10.24 | Утверждён | Марина |

## 1.4. Список рассылки

<https://t.me/+FL4cT2kqwuAxYmEy>

# 2. Общее описание

Проект направлен на создание автоматизированной системы для банка "Глобал Финанс", которая будет выполнять оценку кредитоспособности заемщиков и управление кредитными рисками. Основная цель — автоматизировать анализ данных о клиентах на основе внутренней кредитной истории и финансовых показателей, обеспечив объективное принятие решений по выдаче кредитов. Система будет рассчитывать кредитный скоринг, генерировать отчеты, интегрироваться с существующими ИТ-системами банка и поддерживать работу как с физическими, так и с юридическими лицами.

## 2.1. Терминология

|  |  |
| --- | --- |
| Наименование | Расшифровка |
| Кредитоспособность | Способность заемщика выполнять свои финансовые обязательства по кредиту в срок и в полном объеме. |
| Риско-менеджмент | Процесс управления финансовыми рисками, включающий их оценку, мониторинг и минимизацию. |
| Автоматизация | Процесс использования технологий для выполнения задач без участия человека, что ускоряет процессы и снижает количество ошибок. |
| Платежеспособность | Способность заемщика своевременно выплачивать долг на основании его доходов и расходов. |
| Кредитная история | Информация о прошлых кредитных операциях заемщика, включающая данные о погашении задолженностей, просрочках и текущих обязательствах. |
| Финансовый скоринг | Числовая оценка кредитоспособности заемщика, основанная на его кредитной истории и других финансовых данных. |
| Интерфейс прикладного программирования (API) | Набор инструментов для связи различных программных систем, обеспечивающий обмен данными между ними. |
| Заемщик | Это лицо, которое обращается в банк или финансовую организацию для получения кредита или займа. |
| Финансовые показатели | Ключевые метрики, такие как доходы, активы, рентабельность, используемые для оценки финансового состояния заемщика. |

## 2.2. Соглашения, принятые в документах

Соглашения, принятые в документах, касающихся обязательств сторон, четко регулируют обязанности, права и ответственность каждой стороны. В таких документах всегда подробно изложен предмет соглашения, который определяет основу сделки. Это однозначно включает оказание услуг, передачу имущества или выполнение конкретных работ. Вводная часть документа непременно содержит определения ключевых понятий, что исключает возможность неоднозначного толкования условий.

Обязательства сторон составляют центральный элемент соглашения. Первая сторона обязана выполнить конкретные действия, которые непосредственно связаны с предметом договора. Это включает оказание услуг, поставку товаров или выполнение работ в строгом соответствии с условиями, сроками и требованиями к качеству. Вторая сторона, в свою очередь, обязана принять результаты выполнения этих обязательств и своевременно оплатить их или выполнить другие предусмотренные компенсации. Документ может предусматривать дополнительные обязательства, такие как предоставление необходимых ресурсов или содействие в выполнении основной части договора.

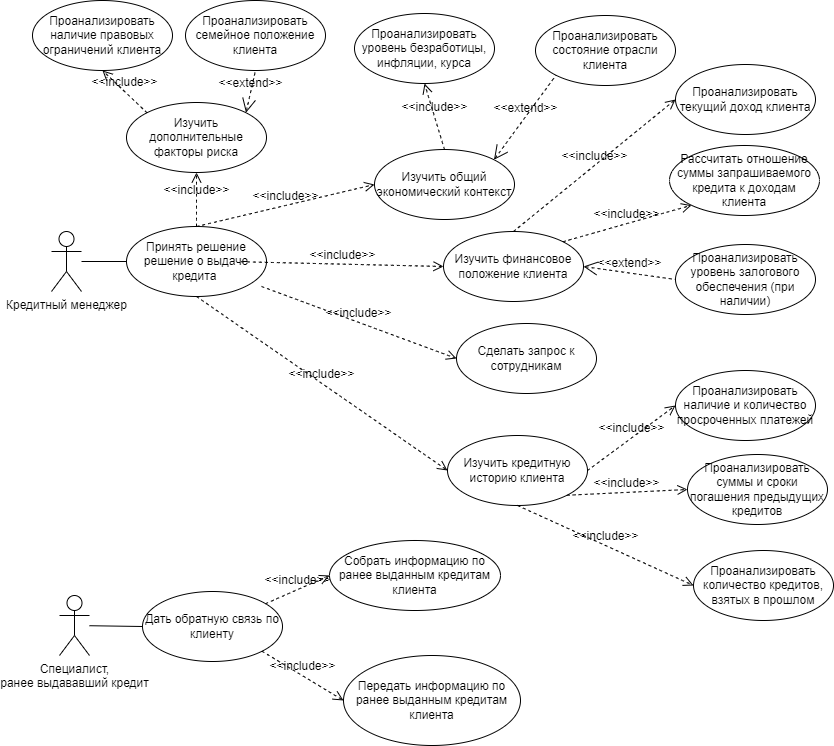
Ответственность сторон за невыполнение или ненадлежащее выполнение обязательств также четко закреплена в документах. В случае нарушения условий договора применяются штрафные санкции или требуется возмещение убытков. В то же время, при наступлении форс-мажорных обстоятельств, таких как природные катастрофы, войны или другие непредвиденные события, стороны освобождаются от ответственности за неисполнение обязательств.

Сроки действия договора и условия его исполнения описаны с точностью. Установлены конкретные даты начала и окончания действия договора, а также предусмотрены четкие условия его расторжения, будь то по взаимному согласию сторон или вследствие невыполнения обязательств одной из них. В документах всегда содержатся положения о конфиденциальности, особенно если сделка подразумевает обмен коммерчески ценной или личной информацией.

Процедура урегулирования споров также строго регламентирована. Стороны заранее договариваются о способах разрешения конфликтов, будь то переговоры, арбитраж или судебное разбирательство. В заключительных положениях указывается порядок внесения изменений в договор, а также содержится контактная информация сторон и их юридические реквизиты. Таким образом, соглашение представляет собой комплексный документ, который надежно защищает интересы всех участников и исключает возможность возникновения недоразумений и споров в процессе его выполнения.

## 2.3. Общий взгляд на продукт, бизнес-процесс

Use-case системы «as is»



## 2.4. Обзор бизнес-требований

1. Повышение точности оценки кредитоспособности: внедрение автоматизированной системы для анализа кредитоспособности клиентов на основе объективных данных. Цель — минимизировать субъективные ошибки и повысить точность оценки.
2. Оптимизация времени обработки заявок: автоматизация обработки заявок на кредиты, что существенно сокращает время проверки информации и принятия решений.
3. Повышение лояльности клиентов: ускорение и повышение точности обработки заявок способствует улучшению клиентского опыта, увеличению доверия и удержанию клиентов.
4. Снижение операционных рисков: автоматизация рутинных процессов снижает вероятность ошибок, связанных с человеческим фактором, и улучшает контроль над операциями.
5. Сокращение затрат на обработку заявок: снижение затрат благодаря уменьшению ручной работы сотрудников и повышению эффективности процессов.
6. Минимизация человеческого фактора: система устраняет зависимость от субъективных решений сотрудников путем автоматизации алгоритмов оценки кредитоспособности.
7. Повышение прозрачности процесса: автоматизация отчетности и аудит предоставляют прозрачность в принятии решений о выдаче кредитов, что позволяет как сотрудникам, так и клиентам видеть причины одобрения или отказа по заявкам.

Эти бизнес-требования направлены на автоматизацию, повышение эффективности и точности работы банка, минимизацию рисков и улучшение клиентского опыта.

## 2.5. Описание текущей системы

Текущая система в банке «Глобал Финанс» основана на ручной оценке кредитоспособности клиентов. Основные проблемы связаны с отсутствием автоматического централизованного анализа и зависимости от субъективного мнения сотрудников. Процесс занимает много времени из-за необходимости собирать информацию вручную, запрашивая ее у различных сотрудников.

### Участники процесса:

1. **Менеджеры банка** – решают, выдавать кредит или нет, основываясь на анализе предыдущих кредитов и обратной связи сотруников.
2. **Финансовый департамент** – занимается отчетностью и расчетами возможных потерь по кредитам.
3. **Аналитики** – участвуют в дальнейшем взаимодействии с системой для обработки и анализа данных.

### Текущий процесс:

Менеджеры изучают кредитную историю клиента, запрашивают информацию у коллег, которые выдали предыдущие кредиты, и принимают решение о выдаче нового кредита. Процесс затягивается из-за длительной оценки информации, которая поступает из разных источников и может быть неполной или устаревшей.

## 2.6. Предлагаемые методы решения

Автоматизация оценки кредитоспособности:

Будет внедрена подсистема, которая будет анализировать кредитоспособность клиентов на основе объективных данных, таких как кредитная история, доходы, активы и задолженности. Это позволит исключить субъективные факторы, улучшив точность и надежность оценки.

Оптимизация процесса обработки заявок:

Автоматизация проверки и обработки заявок на кредиты существенно сократит время обработки, повысив скорость принятия решений и тем самым сократив ожидание клиентов. Это снизит нагрузку на сотрудников и сделает процесс более быстрым.

Уменьшение ручной работы и операционных рисков:

Внедрение автоматизированных алгоритмов и стандартов для проверки данных заемщиков устранит человеческие ошибки, которые могут возникнуть при ручной обработке. Это повысит стабильность и контроль над операциями.

Снижение затрат:

Уменьшение необходимости ручной работы сотрудников, таких как проверка данных, сбор информации и расчет кредитных рисков, приведет к снижению операционных затрат. Сотрудники смогут сосредоточиться на более сложных задачах, требующих их внимания.

Повышение прозрачности и доверия:

Автоматизированная система будет генерировать отчеты, показывающие причины принятия решений по кредитам. Эти отчеты будут доступны для клиентов и сотрудников, что повысит прозрачность процесса и доверие к банку.

Масштабируемость системы:

Проектирование решения с возможностью масштабирования позволит банку эффективно обрабатывать увеличивающиеся объемы данных и число клиентов без потери производительности.

## 2.7. Взаимодействие пользовательских ролей

Список ролей:

|  |  |
| --- | --- |
| Роль | Свойства, возможности и особенности роли |
| *Разработчик (может менять что-либо)* | Имеет полный доступ к системе и может вносить изменения в код, архитектуру и базы данных. Разрабатывает новые функции, устраняет баги и интегрирует внешние источники данных. Тесно взаимодействует с аналитиками для реализации бизнес-логики. |
| *Аналитик (может делать запросы к данным)* | Имеет доступ к данным и может делать запросы для проведения анализа кредитоспособности и оценки рисков. Не может изменять систему, но может формировать отчеты и запросы для исследования данных. Работает с разработчиками для внедрения необходимых изменений. |
| *Менеджер (может делать запросы к данным)* | Имеет доступ к данным и отчетам, может делать запросы для получения информации по кредитным заявкам и рискам. Отвечает за управление процессом кредитования на основе данных. Не может вносить изменения в систему. |
| *Управленцы (будут видеть только конечные результаты об общей оценке)* | Имеют доступ только к итоговым отчетам и общей оценке кредитных рисков. Используют результаты для стратегического планирования и принятия решений. Не взаимодействуют с системой на уровне данных или функциональности. |

## 2.8. Предположения, допущения, зависимости

Проект повысит эффективность обработки кредитных заявок. Автоматизация снижает время обработки заявок и уменьшит количество ошибок, если не будет выявлено значительных проблем с интеграцией данных.

Автоматизация снижает операционные издержки. Система заменит часть рутинной работы сотрудников, что позволит сократить затраты, если не потребуется дополнительных ресурсов для обслуживания.

Решение обеспечит более точную оценку кредитоспособности. Использование машинного обучения и внешних данных повысит точность анализа заемщиков, если алгоритмы будут корректно обучены.

Проект обеспечит соответствие нормативным требованиям. Система будет учитывать действующие законы по защите данных и регулированию кредитования, если не будет выявлено нарушений в обработке персональных данных.

Интеграция с внешними источниками данных будет успешной. Система сможет собирать информацию из кредитных бюро и других внешних источников, если не возникнут технические или юридические ограничения.

Управленцы получают достоверные и актуальные данные для принятия решений. Система будет предоставлять точные и своевременные отчеты, если данные будут обновляться в реальном времени без сбоев.

Проект улучшит клиентский опыт. Ускоренная обработка заявок и более точные решения повысят удовлетворенность клиентов, если система будет работать стабильно.

Масштабируемость системы позволяет обрабатывать большее количество заявок. Решение может расширяться с ростом банка, если архитектура будет достаточно гибкой и оптимизированной.

Система минимизирует влияние человеческого фактора. Автоматизация снизит зависимость от субъективного мнения сотрудников при оценке рисков, если все ключевые аспекты будут учтены в алгоритмах.

Проект будет интегрирован в существующую IT-инфраструктуру банка. Решение должно беспрепятственно работать с текущими системами, если не возникнут несовместимости или технические ограничения.

## 2.9. Границы проекта

Ограничение на бюджет — до 1 миллиарда рублей.

Реализация проекта в течение 1 года с последующей поддержкой и обновлением по необходимости.

Система будет работать на операционных системах Windows и Astra Linux.

Требуется совместимость с существующей ИТ-инфраструктурой банка.

Обработка до 100 ТБ данных, включающих кредитную историю, финансовые показатели и другие метрики заемщиков.

Система должна учитывать нормативные акты и стандарты, связанные с кредитными отношениями в России и странах СНГ.

## 2.10. Ссылки\*

Сделано во внутреннем контуре банка, ссылки отсутствуют.

# 3. Функциональность

*Краткое описание возможностей системы*

Система автоматизации риско-менеджмента и анализа кредитоспособности, являющаяся подсистемой в составе существующей АИС банка “Глобал Финанс”, предназначена для оценки платежеспособности заемщиков и минимизации кредитных рисков. Основной задачей системы является предоставление комплексного и объективного анализа кредитоспособности клиентов (как физических, так и юридических лиц), основываясь на исторических данных и актуальной финансовой информации.

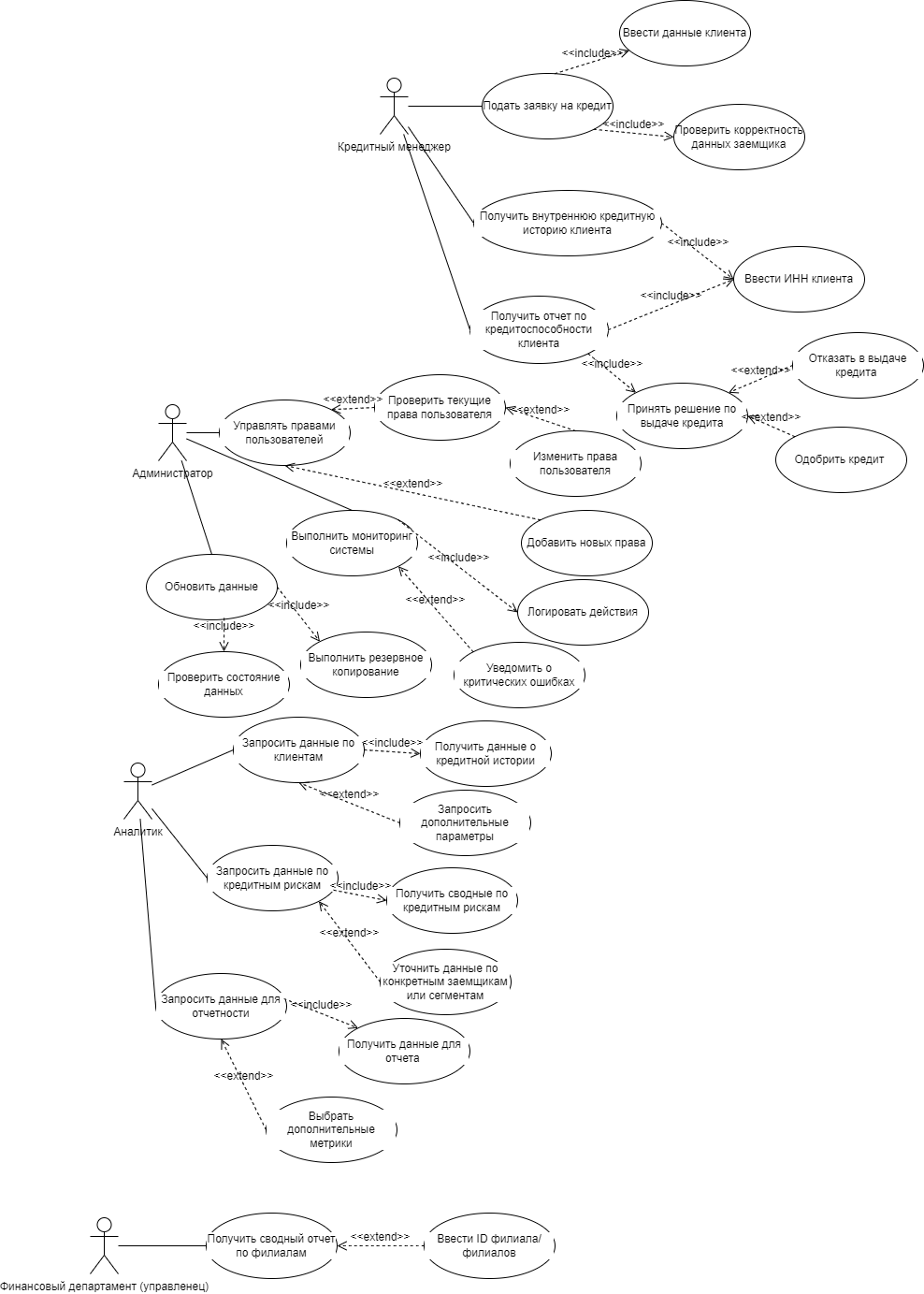
## 3.1. Функциональные требования

Список функциональных требований:

|  |  |
| --- | --- |
| **Наименование** | **Описание** |
| Автоматический сбор данных о заемщике из различных источников | Подсистема автоматически собирает и интегрирует данные о заемщике из внутренних баз данных банка и внешних источников, таких как кредитные бюро и другие организации |
| Анализ финансового состояния | Функция анализа финансовых показателей заемщика, таких как доходы, активы, задолженности и другие важные метрики, чтобы оценить его финансовое положение |
| Автоматический расчет кредитного скоринга | Подсистема автоматически рассчитывает кредитный скоринг заемщика на основе заданных параметров и метрик, минимизируя субъективность в процессе принятия решений |
| Предоставление интерфейса для поиска клиентов по ИНН или ФИО с отображением ключевых метрик | Функциональность для поиска клиентов по ИНН или ФИО с последующим отображением ключевых данных, таких как кредитная история, финансовые показатели и статус заявок |
| Генерация отчетов | Система автоматически генерирует отчеты с ключевыми показателями по филиалам, заемщикам, кредитным рискам и финансовой устойчивости, предоставляя их пользователям в удобной форме |

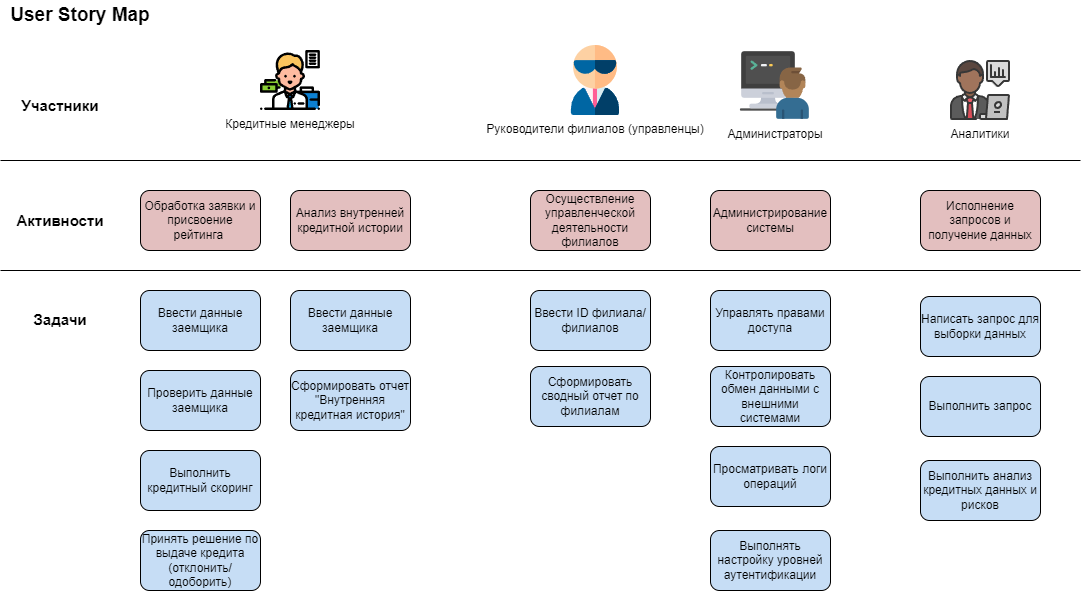
## 3.2. Варианты использования (Use Cases)

Диаграммы Use-cases системы «to be»



## 3.3. Пользовательские истории (User Stories)

Диаграммы User-stories системы «to be»



## 3.4. Внутренние рабочие процессы системы\*

Внутренние рабочие процессы системы автоматизации риско-менеджмента и анализа кредитоспособности включают в себя несколько ключевых этапов, которые обеспечивают точную оценку платежеспособности заемщиков и минимизацию рисков. Каждый из этапов автоматизирован и работает с различными источниками данных, обеспечивая непрерывность и оперативность процесса.

*1. Сбор данных*

Система собирает информацию о клиентах из внутренней базы данных банка (Oracle, в будущем PostgreSQL), которая содержит историю кредитных операций, текущие обязательства, залоги и финансовые показатели клиентов.

Данные обновляются ежедневно через встроенный механизм автоматической синхронизации, который обеспечивает актуальность информации. Новые события (например, оформление кредита, изменение финансовых показателей клиента) заносятся в систему не позднее следующего дня.

Система может быть расширена для работы с внешними источниками, такими как бюро кредитных историй, для получения информации о новых клиентах.

*2. Анализ данных и расчет кредитоспособности*

После поступления данных они проходят проверку на полноту и корректность. Некорректные или неполные данные помечаются для дальнейшего ручного рассмотрения аналитиками.

На основе собранных данных система автоматически рассчитывает ключевые финансовые показатели заемщика (коэффициенты ликвидности, уровень долговой нагрузки, объем залоговых активов). Для юридических лиц дополнительно рассчитываются показатели финансовой устойчивости (например, коэффициенты рентабельности).

Система использует алгоритмы оценки рисков, которые включают предиктивные модели на основе исторических данных. Эти модели помогают предсказать вероятность дефолта клиента и автоматически генерируют показатели кредитных рисков.

На основании скоринговой модели и показателей финансового анализа система присваивает каждому клиенту кредитный рейтинг. Этот рейтинг отражает платежеспособность клиента и уровень риска, связанный с выдачей кредита.

*3. Формирование отчетов и визуализация*

По запросу пользователей (менеджеров, аналитиков, управляющих) система генерирует детализированные отчеты по кредитоспособности клиента. Отчеты включают информацию о кредитной истории, рассчитанных метриках, залогах, финансовых показателях и других ключевых факторах.

Система автоматически создает сводные отчеты, агрегирующие данные по группе клиентов (например, по категориям “физические лица” или “юридические лица”). Эти отчеты помогают управленцам и аналитикам видеть общую картину кредитных рисков.

В рамках дополнительных отчетов система может предоставить информацию о залоговых средствах и коэффициентах финансовой устойчивости для юридических лиц.

Отчеты и аналитические данные представлены в виде визуальных дашбордов с плиточной структурой, что упрощает анализ и восприятие данных. Пользователи могут быстро получить доступ к ключевым показателям через интуитивно понятный интерфейс.

*4. Принятие решений по кредитованию*

Менеджеры и финансовые аналитики используют сгенерированные системой отчеты для оценки заемщика. На основе представленных данных они могут принять решение о выдаче кредита или отказе.

Система интегрирована с существующими процессами банка по подаче заявок на кредит, что позволяет напрямую использовать результаты анализа для формирования заключений по заявкам.

Финансовый департамент может использовать отчеты для анализа возможных рисков и формирования резервов на покрытие потенциальных убытков от невыплаченных кредитов.

*5. Мониторинг и контроль доступа*

Система поддерживает ролевую модель управления доступом. Разработчики имеют полный доступ для внесения изменений, аналитики и менеджеры могут запрашивать данные и генерировать отчеты, а управленцы имеют доступ только к конечным результатам оценки рисков.

Система ведет журнал действий пользователей для отслеживания любых изменений и запросов к данным. Это помогает обеспечить контроль за действиями сотрудников и позволяет своевременно выявлять возможные нарушения или ошибки.

*6. Автоматическая поддержка и обновление системы*

Алгоритмы оценки кредитных рисков и финансового скоринга могут обновляться по мере появления новых данных и изменений во внешней среде (например, изменения макроэкономических условий).

Система разработана с возможностью масштабирования и обновления функционала, что позволяет учитывать изменения в политике банка, новые нормативные акты или требования рынка.

Внутренняя команда аналитиков и технической поддержки контролирует качество данных, исправляет возможные ошибки и предоставляет помощь пользователям системы.

## 3.5. Перечень особенностей («фич») или описания функций

Отличительные возможности системы:

|  |  |
| --- | --- |
| **Наименование “фичи”** | **Описание “фичи”** |
| Интеграция данных из множества источников | Система поддерживает сбор данных как из внутренних баз банка, так и из внешних источников (кредитные бюро, другие организации), что обеспечивает более полное понимание финансового состояния заемщика |
| Реальное время обновления информации | Данные о клиентах обновляются автоматически ежедневно, что гарантирует актуальность информации при принятии решений о кредитах |
| Модуль предсказательного анализа кредитных рисков | Использование предсказательныхой скорирнговой модели для оценки вероятности дефолта заемщика, что помогает минимизировать кредитные риски |
| Интуитивный интерфейс с плиточной визуализацией данных | Удобный для пользователя интерфейс с интерактивными “плитками”, упрощающими навигацию по ключевым показателям, включая кредитные истории и финансовые метрики |
| Многоуровневая система доступа | Различные категории пользователей (менеджеры, аналитики, руководители) имеют доступ к различным уровням данных и функций в зависимости от их ролей, что улучшает безопасность и управляемость |
| Гибкость в настройке отчетов | Пользователи могут легко адаптировать отчеты под свои нужды, используя предустановленные шаблоны и настраиваемые метрики, что делает анализ более гибким и эффективным |
| Поддержка масштабируемости для будущих изменений | Система спроектирована таким образом, что позволяет легко добавлять новые источники данных, улучшать существующие модели и расширять функциональность без необходимости серьезных изменений в архитектуре |
| Мультиплатформенная поддержка | Совместимость с различными операционными системами, включая Windows и Astra Linux, что позволяет использовать систему на разных платформах |
| Модуль аудита и мониторинга активности | Система отслеживает действия пользователей, что позволяет вести журнал всех изменений, обеспечивая прозрачность и контроль за операциями |
| Высокая скорость принятия решений | Благодаря автоматизации оценки кредитоспособности и расчетам скоринга система сокращает время на обработку заявок и принятие решений |

## 3.6. Функциональность администратора

Список функций:

|  |  |
| --- | --- |
| **Наименование функции** | **Описание функции** |
| Управление пользователями | Администратор может создавать, редактировать и удалять учетные записи пользователей, а также назначать и изменять роли и права доступа для различных категорий пользователей (разработчики, аналитики, менеджеры) |
| Настройка прав доступа | Возможность назначения прав доступа к различным данным и функциям системы в зависимости от роли пользователя, что позволяет обеспечить безопасность и контроль над использованием данных |
| Мониторинг и аудит действий пользователей | Администратор имеет доступ к журналам действий пользователей, что позволяет отслеживать все операции в системе, обеспечивая прозрачность и контроль за изменениями данных и функциональности |
| Настройка параметров системы | Администратор может изменять глобальные параметры системы, такие как частота обновления данных, настройки интеграции с внешними API, а также параметры безопасности и уведомлений |
| Резервное копирование и восстановление данных | Управление процессом резервного копирования данных и их восстановление в случае сбоя или утраты информации, что обеспечивает защиту данных |
| Управление интеграцией с внешними системами | Настройка и управление интеграцией системы с внешними источниками данных через API (REST, SOAP) и другими внешними сервисами |
| Обновление системы и управление версиями | Администратор может управлять обновлениями системы, устанавливать новые версии, следить за их совместимостью и при необходимости откатывать изменения |
| Управление базами данных | Администратор имеет доступ к настройкам баз данных, может управлять подключениями к БД, следить за их состоянием и производительностью |
| Управление безопасностью | Настройка параметров безопасности, включая политику паролей, аутентификацию, авторизацию, защиту от несанкционированного доступа, шифрование данных и мониторинг безопасности |
| Мониторинг производительности системы | Администратор может отслеживать производительность системы, загруженность серверов, объемы обработки данных и прочие метрики, что помогает своевременно выявлять узкие места и сбои |
| Настройка отчетов и уведомлений | Возможность настройки параметров генерации отчетов, частоты их обновления, а также настройки автоматических уведомлений для пользователей и администраторов |

## 3.7. Обработка ошибок

Табличка типовых ошибок, о которых может сообщать система:

Коды состояния SQLSTATE

|  |  |  |
| --- | --- | --- |
| **Код** | **Условие** | **Ошибка (ы) Oracle** |
| 00000 | успешное завершение | ORA-00000 |
| 01000 | предупреждение | -- |
| 01001 | конфликт операций с курсором | -- |
| 01002 | ошибка отключения | -- |
| 01003 | НУЛЕВОЕ значение исключено в функции set | -- |
| 01004 | строковые данные -усечение вправо | -- |
| 01005 | недостаточно областей для описания элементов | -- |
| 01006 | привилегия не отозвана | -- |
| 01007 | привилегия не предоставлена | -- |
| 01008 | неявное заполнение нулевыми битами | -- |
| 01009 | условие поиска слишком длительного для информационной схемы | -- |
| 0100A | выражение запроса слишком длинное для информационной схемы | -- |
| 02000 | нет данных | ORA-01095  ORA-01403 |
| 07000 | динамическая ошибка SQL | -- |
| 07001 | предложение using не соответствует спецификациям параметров | -- |
| 07002 | предложение using не соответствует целевым спецификациям | -- |
| 07003 | спецификация курсора не может быть выполнена | -- |
| 07004 | предложение using, необходимое для динамических параметров | -- |
| 07005 | подготовленная инструкция, а не спецификация курсора | -- |
| 07006 | нарушение атрибута ограниченного типа данных | -- |
| 07007 | использование предложения, требуемого для компонентов результата, недопустимое количество дескрипторов | -- |
| 07008 | недопустимое количество дескрипторов | SQL-02126 |
| 07009 | недопустимый индекс дескриптора | -- |
| 08000 | исключение подключения | -- |
| 08001 | SQL-клиенту не удается установить SQL-соединение | -- |
| 08002 | имя соединения - use | -- |
| 08003 | соединение не существует | SQL-02121 |
| 08004 | SQL-сервер отклонил SQL-соединение | -- |
| 08006 | сбой подключения | -- |
| 08007 | разрешение транзакции неизвестно | -- |
| 0A000 | функция не поддерживается | ORA-03000..03099 |
| 0A001 | множественные серверные транзакции | -- |
| 21000 | нарушение количества элементов | ORA-01427  SQL-02112 |
| 22000 | исключение данных | -- |
| 22001 | строковые данные - правильное усечение | ORA-01406 |
| 22002 | НУЛЕВОЕ значение - нет параметра индикатора | SQL-02124 |
| 22003 | числовое значение вне диапазона | ORA-01426 |
| 22005 | ошибка в назначении | -- |
| 22007 | неверный формат даты и времени | -- |
| 22008 | переполнение поля datetime | ORA-01800..01899 |
| 22009 | недопустимое значение смещения часового пояса | -- |
| 22011 | ошибка подстроки | -- |
| 22012 | деление на ноль | ORA-01476 |
| 22015 | переполнение поля интервала | -- |
| 22018 | недопустимое символьное значение для приведения | -- |
| 22019 | недопустимый управляющий символ | ORA-00911 |
| 22021 | персонажа нет в репертуаре | -- |
| 22022 | переполнение индикатора | ORA-01411 |
| 22023 | недопустимое значение параметра | ORA-01025  ORA-04000..04019 |
| 22024 | незавершенная строка C | ORA-01479  ORA-01480 |
| 22025 | недопустимая управляющая последовательность | ORA-01424  ORA-01425 |
| 22026 | string data-length mismatch | ORA-01401 |
| 22027 | trim error | - |
| 23000 | integrity constraint violation | ORA-1400, ORA-02290..02299 |
| 24000 | invalid cursor state | ORA-001002  ORA-001003  SQL-02114  SQL-02117 |
| 25000 | invalid transaction state | SQL-02118 |
| 26000 | неверное имя инструкции SQL | -- |
| 27000 | нарушение, вызванное изменением данных | -- |
| 28000 | неверная спецификация авторизации | -- |
| 2A000 | прямая синтаксическая ошибка SQL или нарушение правил доступа | -- |
| 2B000 | зависимые дескрипторы привилегий все еще существуют | -- |
| 2C000 | недопустимое имя набора символов | -- |
| 2D000 | недопустимое завершение транзакции | -- |
| 2E000 | неверное имя подключения | -- |
| 33000 | неверное имя SQL-дескриптора | -- |
| 34000 | неверное имя курсора | -- |
| 35000 | недопустимый номер условия | -- |
| 37000 | ошибка динамического синтаксиса SQL или нарушение правил доступа | -- |
| 3C000 | неоднозначное имя курсора | -- |
| 3D000 | неверное имя каталога | -- |
| 3F000 | неверное имя схемы | -- |
| 40000 | откат транзакции | ORA-02091  ORA-02092 |
| 40001 | сбой сериализации | -- |
| 40002 | нарушение ограничения целостности | -- |
| 40003 | неизвестное завершение инструкции | -- |
| 42000 | синтаксическая ошибка или нарушение правил доступа | ORA-00022  ORA-00251  ORA-00900..00999  ORA-01031  ORA-01490..01493  ORA-01700..01799  ORA-01900..02099  ORA-02140..02289  Сейчас-02420..02424  Сейчас-02450..02499  Сейчас-03276..03299  Сейчас-04040..04059  Сейчас-04070..04099 |
| 44000 | с нарушением опции проверки | НОМЕР-01402 |
| 60000 | системная ошибка | НОМЕР-00370..00429  НОМЕР-00600..00899  НОМЕР-06430..06449  НОМЕР-07200..07999  НОМЕР-09700..09999 |
| 61000 | общий сервер был отключен из-за ошибок процесса | НОМЕР-00018..00035  НОМЕР-00050..00068  НОМЕР-02376..02399  НОМЕР-04020..04039 |
| 62000 | общий сервер был отключен из-за ошибок процесса | НОМЕР-00100..00120  НОМЕР-00440..00569 |
| 63000 | Oracle \* X имеет место ошибка интерфейса с двумя задачами | НОМЕР-00150..00159  НОМЕР-02700..02899  НОМЕР-03100..03199  ORA-06200..06249  SQL-02128 |
| 64000 | ошибки в файлах управления, файлах баз данных и файлах повторных операций; ошибки при архивировании и восстановлении данных | ORA-00200..00369  ORA-01100..01250 |
| 65000 | Ошибки PL / SQL | НОМЕР-06500..06599 |
| 66000 | Ошибки драйвера Oracle Net | НОМЕР-06000..06149  НОМЕР-06250..06429  НОМЕР-06600..06999  НОМЕР-12100..12299  НОМЕР-12500..12599 |
| 67000 | ошибки лицензирования | НОМЕР-00430..00439 |
| 69000 | Ошибки SQL \* Connect | НОМЕР-00570..00599  НОМЕР-07000..07199 |
| 72000 | Ошибки фазы выполнения SQL | НОМЕР-00001  НОМЕР-01000..01099  НОМЕР-01401..01489  НОМЕР-01495..01499  НОМЕР-01500..01699  Сейчас-02400..02419  Сейчас-02425..02449  Сейчас-04060..04069  Сейчас-08000..08190  Сейчас-12000..12019  Сейчас-12300..12499  Сейчас-12700..21999 |
| 82100 | out of memory (could not allocate) | SQL-02100 |
| 82101 | inconsistent cursor cache (UCE/CUC mismatch) | SQL-02101 |
| 82102 | несогласованный кэш курсора (нет записи CUC для UCE) | SQL-02102 |
| 82103 | несогласованный кэш курсора (ссылка CUC вне диапазона) | SQL-02103 |
| 82104 | несогласованный кэш курсора (нет доступного CUC) | SQL-02104 |
| 82105 | несогласованный кэш курсора (нет записи CUC в кэше) | SQL-02105 |
| 82106 | несогласованный кэш курсора (недопустимый номер курсора) | SQL-02106 |
| 82107 | программа слишком старая для библиотеки времени выполнения; перекомпилировать | SQL-02107 |
| 82108 | недопустимый дескриптор, переданный в библиотеку времени выполнения | SQL-02108 |
| 82109 | несогласованный кэш хоста (ошибка выхода за пределы диапазона) | SQL-02109 |
| 82110 | несогласованный кэш хоста (недопустимый тип SQL) | SQL-02110 |
| 82111 | ошибка согласованности кучи | SQL-02111 |
| 82113 | сбой внутренней согласованности генерации кода | SQL-02115 |
| 82114 | генератор реентерабельного кода выдал недопустимый контекст | SQL-02116 |
| 82117 | недопустимое ОТКРЫТИЕ или ПОДГОТОВКА к этому подключению | SQL-02122 |
| 82118 | контекст приложения не найден | SQL-02123 |
| 82119 | не удалось получить текст сообщения об ошибке | SQL-02125 |
| 82120 | Несоответствие версий прекомпилятора / SQLLIB | SQL-02127 |
| 82121 | Ошибка NCHAR; выбранное количество байтов нечетное | SQL-02129 |
| 82122 | Интерфейс EXEC TOOLS недоступен | SQL-02130 |
| 82123 | используемый контекст времени выполнения | SQL-02131 |
| 82124 | не удалось выделить контекст времени выполнения | SQL-02132 |
| 82125 | не удалось инициализировать процесс для использования с потоками | SQL-02133 |
| 82126 | недопустимый контекст времени выполнения | SQL-02134 |
| HZ000 | удаленный доступ к базе данных | -- |

Коды состояния HRESULT

|  |  |
| --- | --- |
| **Код** | **Условие** |
| 0x00000000 | S\_OK — Успешное выполнение операции |
| 0x80004005 | E\_FAIL — Неизвестная ошибка |
| 0x80070005 | E\_ACCESSDENIED — Доступ запрещен |
| 0x80070057 | E\_INVALIDARG — Неверный аргумент |
| 0x8007045D | E\_IO — Ошибка ввода-вывода устройства |

Коды состояния POSIX (Unix/Linux)

|  |  |
| --- | --- |
| **Код** | **Условие** |
| EPERM | S\_OK — Успешное выполнение операции |
| ENOENT | Нет такого файла или директории |
| ESRCH | Процесс не найден |
| EINTR | Прерывание системного вызова |
| ENOMEM | Недостаточно памяти |

## 3.8. Взаимодействие продукта (с другими продуктами и компонентами) \*

*1. Взаимодействие с внутренними системами*

Продукт будет интегрирован с основными внутренними системами банка. Взаимодействие будет происходить с базой данных (Oracle/PostgreSQL), где будет храниться информация о заемщиках, кредитах и финансовых показателях. Система будет выполнять запросы к базе данных для получения и сохранения необходимой информации.

Кроме того, продукт будет интегрирован с системой бизнес-анализа (например, Tableau), что позволит создавать отчеты и визуализировать данные о кредитоспособности заемщиков и кредитных рисках. Подсистема будет работать как часть существующей автоматизированной информационной системы (АИС) банка, обмениваясь данными с другими модулями.

Также важным аспектом является взаимодействие с системой управления пользователями, которая обеспечит аутентификацию и авторизацию, позволяя управлять доступом к различным уровням информации в соответствии с ролями пользователей.

*2. Взаимодействие с внешними системами*

Система будет осуществлять автоматический обмен данными с внешними бюро кредитных историй, что позволит получать кредитные истории заемщиков и проверять их кредитоспособность. Это взаимодействие будет осуществляться через REST или SOAP API для доступа к данным.

Кроме того, возможно подключение к сторонним финансовым и аналитическим сервисам, которые могут предоставить дополнительные метрики о заемщиках.

*3. Взаимодействие с пользователями*

Система будет ориентирована на различные категории пользователей. Менеджеры банка будут взаимодействовать с системой через удобный пользовательский интерфейс для оценки кредитоспособности заемщиков и принятия решений по заявкам на кредиты. Аналитики будут использовать систему для проведения глубокого финансового анализа и генерации отчетов по кредитным рискам. Финансовый департамент сможет извлекать информацию из системы для получения отчетов о кредитных рисках и формирования резервов.

*4. Взаимодействие с инфраструктурой*

Продукт будет хоститься на серверах приложений, которые обеспечат выполнение бизнес-логики и обработку запросов пользователей. Для обеспечения безопасности и целостности данных будут регулярно создаваться резервные копии, а также система будет интегрирована с системами мониторинга, что позволит отслеживать производительность системы и выявлять потенциальные проблемы.

# 4. Требования к данным

*Форматы загрузки данных*

CSV: Для импорта табличных данных (заемщики, кредиты).

Excel (XLSX): Для загрузки отчетов и небольших объемов данных.

JSON: Для передачи данных через API.

XML: Для обмена структурированными данными.

SQL: Запросы для загрузки в базы данных.

*Форматы выгрузки данных*

CSV: Для экспорта отчетов.

Excel (XLSX): Для отчетов с графиками.

PDF: Для финальных отчетов.

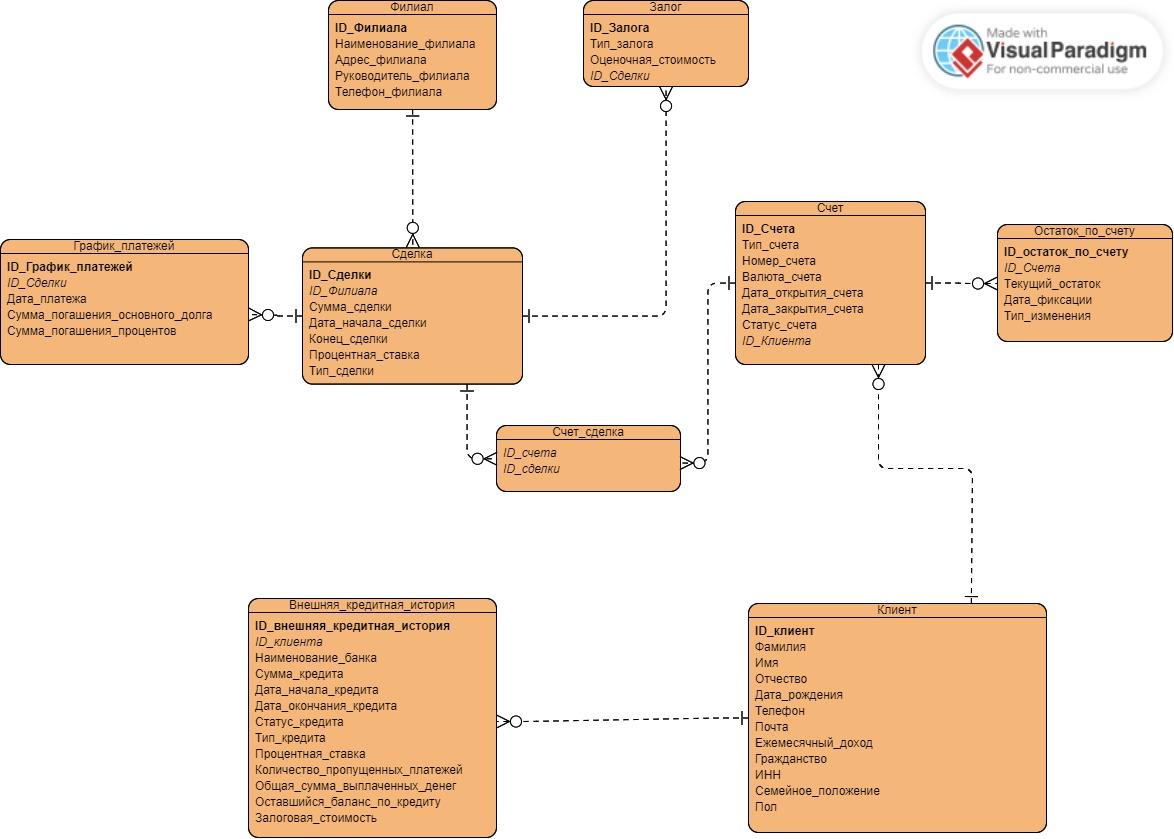
JSON: Для передачи через API.

XML: Для структурированного экспорта.

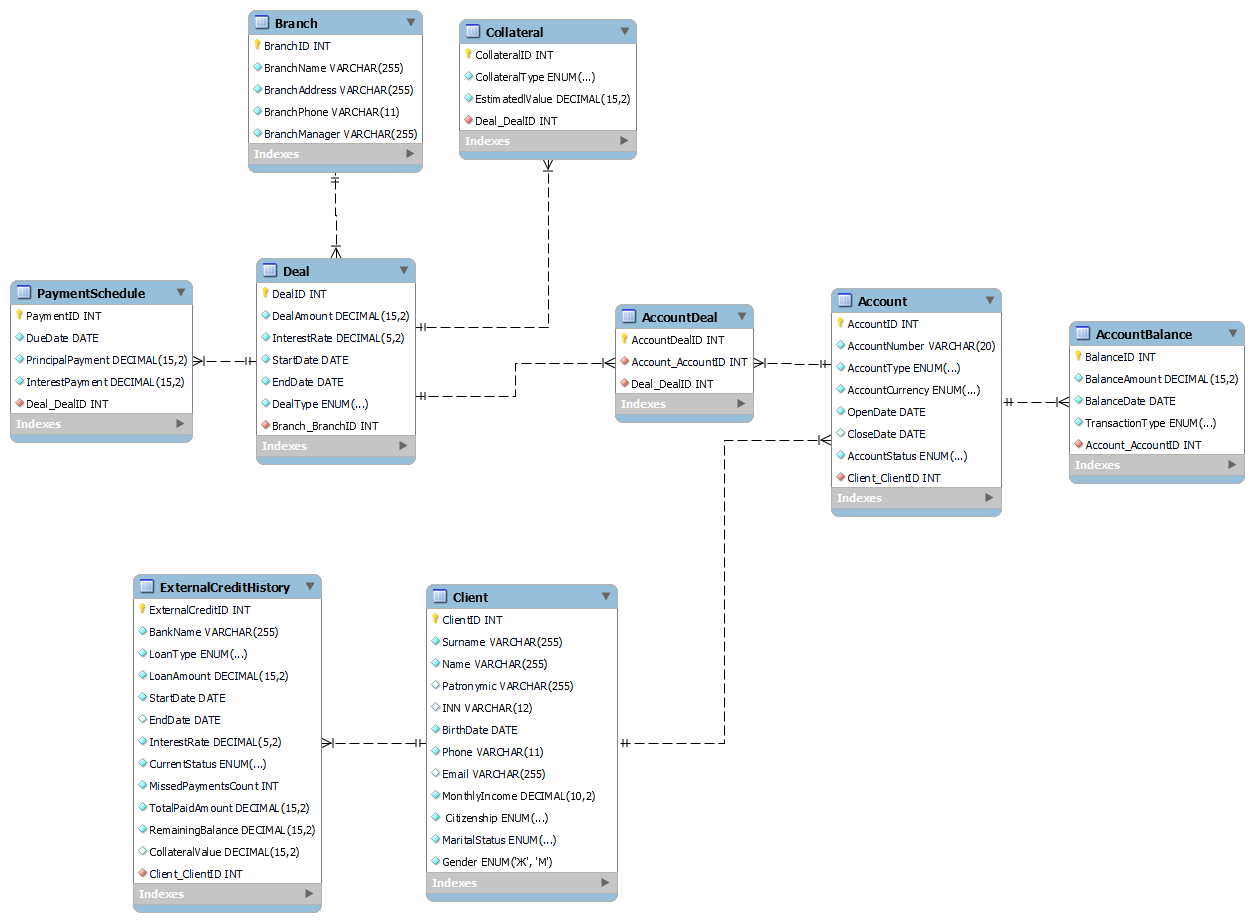
## 4.1. Модели данных

1. ER-диаграммы

Логическая модель данных:



Физическая модель данных, выполненная под СУБД MySQL:



2) Табличное представление

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Таблица** | **Атрибут** | **Тип данных** | **PK** | **FK** |
| Client | ClientID | NUMBER | PK |  |
| Surname | VARCHAR(255) |  |  |
| Name | VARCHAR(255) |  |  |
| Patronymic | VARCHAR(255) |  |  |
| INN | VARCHAR(12) |  |  |
| BirthDate | DATE |  |  |
| Phone | VARCHAR(11) |  |  |
| Email | VARCHAR(255) |  |  |
| MonthlyIncome | DECIMAL(10,2) |  |  |
| Cityzenship | VARCHAR(50) |  |  |
| MaritalStatus | VARCHAR(20) |  |  |
| Gender | VARCHAR(20) |  |  |
| ExternalCreditHistory | ExternalCreditID | NUMBER | PK |  |
| BankName | VARCHAR(255) |  |  |
| LoanType | VARCHAR(100) |  |  |
| LoanAmount | DECIMAL(15,2) |  |  |
| StartDate | DATE |  |  |
| EndDate | DATE |  |  |
| InterestRate | DECIMAL(5,2) |  |  |
| CurrentStatus | VARCHAR(50) |  |  |
| MissedPaymentsCount | NUMBER |  |  |
| TotalPaidAmount | DECIMAL(15,2) |  |  |
| RemainingBalance | DECIMAL(15,2) |  |  |
| CollateralValue | DECIMAL(10,2) |  |  |
| Client\_ClientID | NUMBER |  | FK |
| Branch | BranchID | NUMBER | PK |  |
| Branch\_Name | VARCHAR(255) |  |  |
| Branch\_Address | VARCHAR(255) |  |  |
| Branch\_Phone | VARCHAR(20) |  |  |
| Branch\_Manager | VARCHAR(255) |  |  |
| PaymentSchedule | PaymentID | NUMBER | PK |  |
| DueDate | DATE |  |  |
| PrincpalPayment | DECIMAL(15,2) |  |  |
| InterestPayment | DECIMAL(15,2) |  |  |
| Deal\_DealID | NUMBER |  | FK |
| Deal | DealID | NUMBER | PK |  |
| DealAmount | DECIMAL(15,2) |  |  |
| InterestRate | DECIMAL(5,2) |  |  |
| StartDate | DATE |  |  |
| EndDate | DATE |  |  |
| DealType | VARCHAR(100) |  |  |
| Branch\_BranchID | NUMBER |  | FK |
| Collateral | CollateralID | NUMBER | PK |  |
| CollateralType | VARCHAR(100) |  |  |
| EstimatedValue | DECIMAL(15,2) |  |  |
| Deal\_DealID | NUMBER |  | FK |
| Account | AccountID | NUMBER | PK |  |
| AccountType | VARCHAR(50) |  |  |
| AccountNumber | VARCHAR(20) |  |  |
| AccountCurrency | VARCHAR(3) |  |  |
| OpenDate | DATE |  |  |
| CloseDate | DATE |  |  |
| AccountStatus | VARCHAR(50) |  |  |
| Client\_ClientID | NUMBER |  | FK |
| AccountDeal | AccountDealID | NUMBER | PK |  |
| Account\_AccountID | NUMBER |  | FK |
| Deal\_Deal\_ID | NUMBER |  | FK |
| AccountBalance | BalanceID | NUMBER | PK |  |
| BalanceAmount | DECIMAL(15,2) |  |  |
| BalanceDate | DATE |  |  |
| TransactionType | VARCHAR(20) |  |  |
| Account\_AccountID | NUMBER |  | FK |

## 4.2. Словарь данных

Маппинг атрибутов системы (по витрине )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Атрибут** | **Алгоритм заполнения** | **Название атрибута** | **Тип данных** | **Nullable** |
| **Витрина “Отчет по филиалам”** | | | | |
| cc.BRANCHID | cc.BRANCHID | Идентификатор филиала | NUMBER | NO |
| cc.BRANCH\_NAME | cc.BRANCH\_NAME | Наименование филиала | VARCHAR(255) | NO |
| cc.BRANCH\_ADDRESS | cc.BRANCH\_ADDRESS | Адрес филиала | VARCHAR(255) | NO |
| cc.COUNT\_CLIENT | Вычисляется в подзапросе WITH CountClient как COUNT( DISTINCT c.CLIENTID). Этот подзапрос объединяет таблицы BRANCH, DEAL, ACCOUNT, CLIENT для вычисления уникального количества клиентов по каждому филиалу. | Количество клиентов | NUMBER | NO |
| cd.Count\_Completed\_Deals | Вычисляется в подзапросе WITH CountCompletedDeals как COUNT(d.DEALID). Объединяет таблицы BRANCH и DEAL, фильтруя сделки с датой окончания меньше текущей даты (SYSDATE). Затем для каждого филиала подсчитывается количество завершенных сделок. | Количество завершенных сделок всех типов | NUMBER | NO |
| sd.Sum\_Completed\_Deals | Вычисляется в подзапросе WITH SumCompletedDeals как SUM(d.DEALAMOUNT). Этот подзапрос суммирует общие суммы завершенных сделок, фильтруя по типу сделки (DEALTYPE = 'Кредитные операции') и датам окончания до текущей даты (d.enddate < SYSDATE). | Сумма завершенных сделок типа кредитные операции | DECIMAL(15,2) | NO |
| cad.COUNT\_ACTIVE\_DEAL | Вычисляется в подзапросе WITH CountActiveDeal как COUNT(d.DEALID). Объединяет таблицы BRANCH и DEAL, фильтруя сделки с датой окончания, которая больше или равна текущей дате (d.enddate >= SYSDATE). Считает активные сделки для каждого филиала. | Количество активных сделок всех типов | NUMBER | NO |
| tal.SUM\_ACTIVE\_LOAN | Вычисляется в подзапросе WITH TotalAmountOfActiveLoans как SUM(d.DEALAMOUNT). Этот подзапрос объединяет таблицы BRANCH и DEAL, фильтруя только сделки типа "Кредитные операции" с датой окончания больше или равной текущей дате. Суммирует общий объем активных кредитных сделок по филиалам. | Общий объем активных сделок типа кредитные операции | DECIMAL(15,2) | NO |
| col.Count\_Overdue\_Loans | Вычисляется в подзапросе WITH CountOverdueLoans как COUNT(DISTINCT cp.DEAL\_DEALID). Подсчитывается количество уникальных сделок, по которым имеются просроченные платежи, путем сравнения данных из подзапроса comparisonpayments, где вычисляется соответствие планируемых и фактических платежей. | Кол-во просроченных сделок типа кредитные операции | NUMBER | NO |
| cop.Count\_Overdue\_payments | Вычисляется в подзапросе WITH CountOverduePayments как COUNT(CP.comparison), где сравниваются запланированные платежи с фактическими. Если планируемый платеж не совпадает с фактическим (или его нет на дату), это считается просроченным платежом. | Кол-во просроченных платежей по сделкам типа кредитные операции | NUMBER | NO |
| sop.sum\_overdue\_payments | Вычисляется в подзапросе WITH SumOverduePayments как SUM(cp.PRINCPALPAYMENT + cp.INTERESTPAYMENT), суммируя все просроченные платежи (и основной долг, и проценты) для каждой сделки, по которым имеются просроченные платежи. | Сумма просроченных сделок типа кредитные операции | DECIMAL(15,2) | NO |
| ii.sum\_interest\_payments | Вычисляется в подзапросе WITH InterestIncome как SUM(cp.INTERESTPAYMENT). Этот подзапрос суммирует только процентные платежи по сделкам типа кредитные операции, где планируемые и фактические платежи совпадают, и сделка активна (дата платежа не прошла). | Доход от процентов по сделкам типа кредитные операции | DECIMAL(15,2) | NO |
|  | Используется системная дата SYSDATE. Хранит дату, на момент которой был сформирован отчет, и используется для дальнейших вычислений при сравнении дат сделок, платежей и других временных атрибутов. | Дата формирования отчета | DATE | NO |
| **Витрина “Кредитный скоринг”** | | | | |
| c.clientid | SELECT DISTINCT c.clientid | Идентификатор клиента | NUMBER | NO |
| ages.age\_score | Расчёт производится на основе WITH AgeScoring | Баллы возраста | NUMBER | NO |
| dtis.dti\_score | Расчёт производится на основе WITH DTIScoring | Баллы DTI | NUMBER | NO |
| ms.marital\_score | Расчёт производится на основе WITH MaritalScoring | Баллы семейного положения | NUMBER | NO |
| cs.CollateralScore | Расчёт производится на основе WITH CollateralScoring | Баллы процента залога от суммы кредита | NUMBER | NO |
| os.PaymentHistoryScore | Расчёт производится на основе WITH OverdueScoring | Баллы просрочек по кредитам | NUMBER | NO |
| rs.TotalLoanScore | Расчёт производится на основе WITH RepaymentScoring | Баллы по сумме ранее взятых кредитов и их погашении | NUMBER | NO |
| acs.ACTIVE\_LOANS\_SCOR | Расчёт производится на основе WITH ActiveCreditsScoring | Баллы по количеству активных кредито | NUMBER | NO |
| mis.INCOME\_SCORE | Расчёт производится на основе WITH MonthlyIncomeScoring | Баллы ежемесячного дохода | NUMBER | NO |
| Сумма баллов | ages.age\_score +  dtis.dti\_score +  ms.marital\_score +  COALESCE(cs.CollateralScore,0) +  os.PaymentHistoryScore +  COALESCE(rs.TotalLoanScore,0) +  COALESCE(acs.ACTIVE\_LOANS\_SCOR,0) +  mis.INCOME\_SCORE | Сумма баллов | NUMBER | NO |
| Категория качества кредита | CASE  WHEN ages.age\_score + dtis.dti\_score + ms.marital\_score + COALESCE(cs.CollateralScore,0) + os.PaymentHistoryScore + COALESCE(rs.TotalLoanScore,0) + COALESCE(acs.ACTIVE\_LOANS\_SCOR,0) + mis.INCOME\_SCORE BETWEEN 1 AND 8 THEN 5  WHEN ages.age\_score + dtis.dti\_score + ms.marital\_score + COALESCE(cs.CollateralScore,0) + os.PaymentHistoryScore + COALESCE(rs.TotalLoanScore,0) + COALESCE(acs.ACTIVE\_LOANS\_SCOR,0) + mis.INCOME\_SCORE BETWEEN 9 AND 16 THEN 4  WHEN ages.age\_score + dtis.dti\_score + ms.marital\_score + COALESCE(cs.CollateralScore,0) + os.PaymentHistoryScore + COALESCE(rs.TotalLoanScore,0) + COALESCE(acs.ACTIVE\_LOANS\_SCOR,0) + mis.INCOME\_SCORE BETWEEN 17 AND 24 THEN 3  WHEN ages.age\_score + dtis.dti\_score + ms.marital\_score + COALESCE(cs.CollateralScore,0) + os.PaymentHistoryScore + COALESCE(rs.TotalLoanScore,0) + COALESCE(acs.ACTIVE\_LOANS\_SCOR,0) + mis.INCOME\_SCORE BETWEEN 25 AND 32 THEN 2  WHEN ages.age\_score + dtis.dti\_score + ms.marital\_score + COALESCE(cs.CollateralScore,0) + os.PaymentHistoryScore + COALESCE(rs.TotalLoanScore,0) + COALESCE(acs.ACTIVE\_LOANS\_SCOR,0) + mis.INCOME\_SCORE BETWEEN 33 AND 40 THEN 1  END AS | Категория качества кредита | NUMBER | NO |
| **Витрина “Внутренняя кредитная история”** | | | | |
| ca.CLIENTID | SELECT ca.CLIENTID AS "ID\_Клиента" | ID\_Клиента | NUMBER | NO |
| ca.BRANCHID | SELECT ca.BRANCHID AS "ID\_Филиала" | ID\_Филиала | NUMBER | NO |
| ca.STARTDATE | SELECT ca.STARTDATE AS "Дата начала сделки" | Дата начала сделки | DATE | NO |
| ca.ENDDATE | SELECT ca.ENDDATE AS "Дата окончания сделки" | Дата окончания сделки | DATE | NO |
| ca.ACCOUNT\_COUNT | Расчёт на основе WITH CountAccounts:  SELECT COUNT(a.ACCOUNTID) для счетов типа 'Счёт основного долга', 'Счёт процентов' | Количество счетов (Счёт основного долга и Счёт процентов) | NUMBER | NO |
| ps.TOTAL\_PAYMENT | Расчёт на основе WITH PaymentSum:  SUM(ps.INTERESTPAYMENT + ps.PRINCPALPAYMENT) для клиентских сделок | Сумма платежа (проценты + основная сумма) | NUMBER | NO |
| mp.MONTHLY\_PAYMENT | Расчёт на основе WITH MonthlyPayment:  SUM(ps.INTERESTPAYMENT + ps.PRINCPALPAYMENT) / COUNT(DISTINCT EXTRACT(MONTH FROM ps.DUEDATE)), если имеются записи платежей, иначе 0. | Месячный платёж | NUMBER | NO |
| op.HAS\_OVERDUE | Определение просрочки на основе WITH OverduePayments:  CASE WHEN COUNT(CASE WHEN ps.DUEDATE < SYSDATE AND (ps.INTERESTPAYMENT > 0 OR ps.PRINCPALPAYMENT > 0) THEN 1 END) > 0 THEN 1 ELSE 0 END | Наличие просрочки (1 - есть, 0 - нет) | NUMBER | NO |

## 4.3. Отчеты

Список отчетов, выводимых системой:

|  |  |
| --- | --- |
| **Отчет** | **Описание** |
| Отчет по филиалам | Отчет, отображающий статистику и данные по кредитным сделкам в разрезе филиалов, включая суммы платежей, количество счетов и просроченные сделки. |
| Кредитный скоринг | Анализ кредитоспособности клиентов на основе различных параметров, таких как возраст, семейное положение, показатель DTI, ежемесячный доход, наличие собственности, кредитная история (по физ. лицу). |
| Внутренняя кредитная история | Внутренняя аналитика кредитной истории клиентов, включающая информацию о прошлых просрочках, суммах платежей и кредитных сделках за все время сотрудничества с клиентом. |

## 4.4. Получение

Источники данных для системы: хранилище данных банка, бюро кредитных историй, бюро финансового мониторинга, платформы для анализа поведения.

## 4.5. Целостность

Для обеспечения целостности данных и защиты их от неправомерного изменения применяются различные методы и технологии:

1. **Контроль доступа** — ограничение прав на изменение данных только для авторизованных пользователей.
2. **Шифрование** — данные шифруются при передаче и хранении, предотвращая их подмену.
3. **Журналы аудита** — запись всех действий по изменению данных для их отслеживания.
4. **Транзакции** — обеспечивают целостность операций, предотвращая частичное внесение данных.
5. **Многофакторная аутентификация (MFA)** — усиливает контроль доступа к данным.
6. **Резервное копирование** — позволяет восстановить данные в случае их изменения или повреждения.

## 4.6. Хранение и утилизация данных

Хранение и утилизация данных в системе будет осуществляться с соблюдением высоких стандартов безопасности и конфиденциальности. Данные о заемщиках, кредитных историях и финансовых метриках будут храниться в реляционных базах данных, что обеспечит быстрый доступ и обработку информации. Все данные будут шифроваться как при хранении, так и при передаче, что защитит конфиденциальную информацию от несанкционированного доступа с использованием современных алгоритмов шифрования, соответствующих отраслевым стандартам. Регулярная система резервного копирования данных будет включать автоматизированные процессы создания резервных копий в защищенных удаленных хранилищах, что обеспечит восстановление данных в случае их утраты. Доступ к данным будет ограничен с помощью системы управления доступом, основанной на ролях (RBAC), позволяя только авторизованным пользователям с соответствующими правами просматривать и изменять данные.

В отношении утилизации данных будет разработана четкая политика, определяющая сроки хранения и методы удаления данных, которые больше не подлежат дальнейшему хранению. Данные будут безопасно удаляться с использованием методов, исключающих возможность восстановления, таких как перезапись данных и физическое уничтожение носителей, когда они больше не нужны. Все действия по утилизации данных будут производиться в соответствии с требованиями законодательства о защите персональных данных и другими актуальными стандартами для обеспечения соблюдения норм. Кроме того, все процессы утилизации будут документироваться для обеспечения прозрачности и отчетности, что позволит отслеживать, какие данные были удалены и на каких основаниях.

# 5. Требования к внешним интерфейсам

Требования заказчика к интерфейсу:

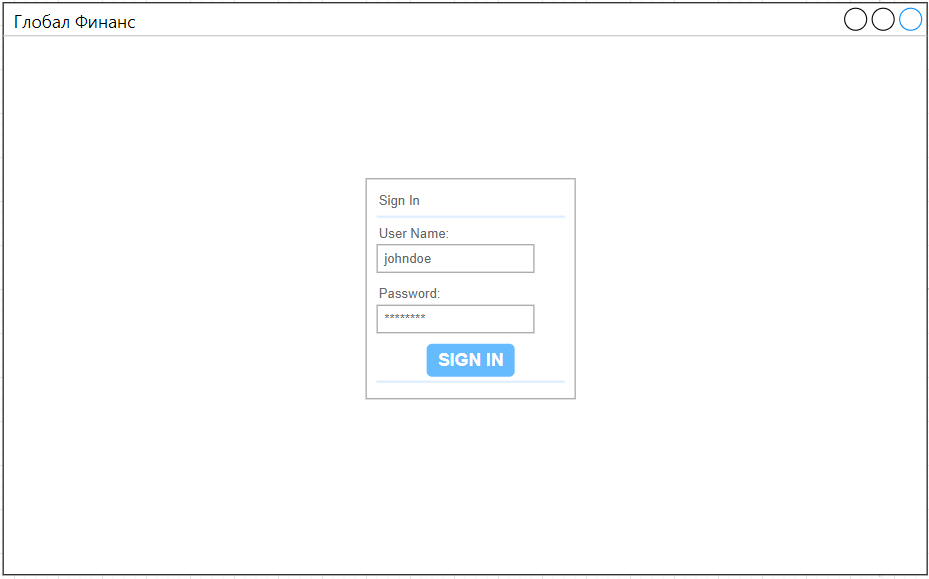
|  |  |
| --- | --- |
| user friendly интерфейс | Заказчик хотел бы видеть интерфейс понятным при использование. |

## 5.1. Пользовательские интерфейсы

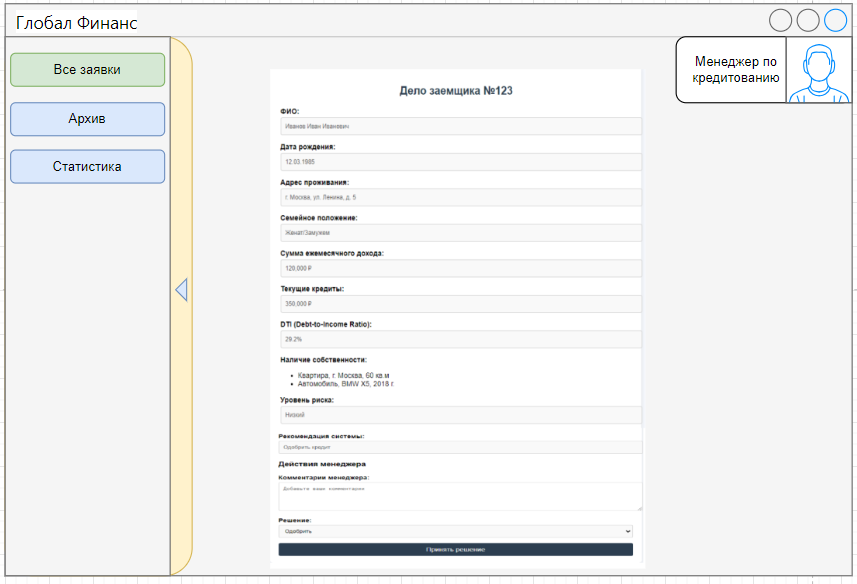
Пользовательский интерфейс отсутствует

### 5.1.1. Прототипы дизайнов

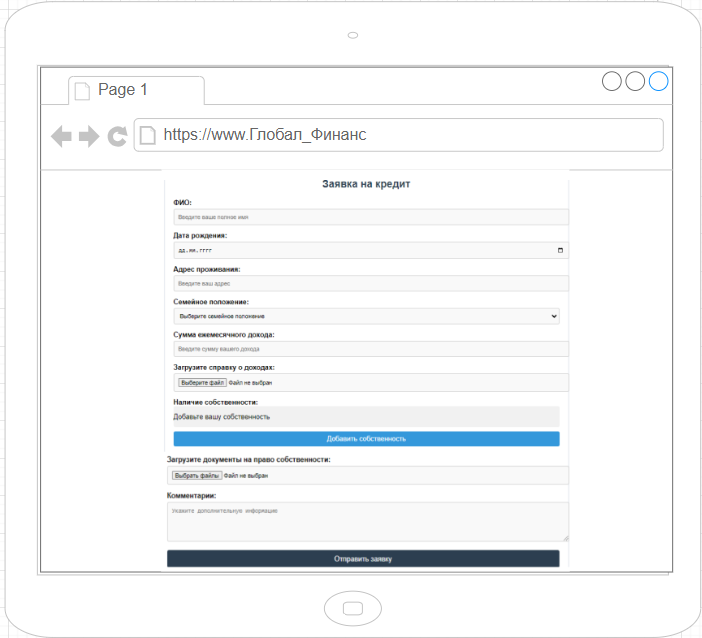
Первая экранная форма “Авторизация пользователя “. Такую форму видит только пользователь системы (работник банка)



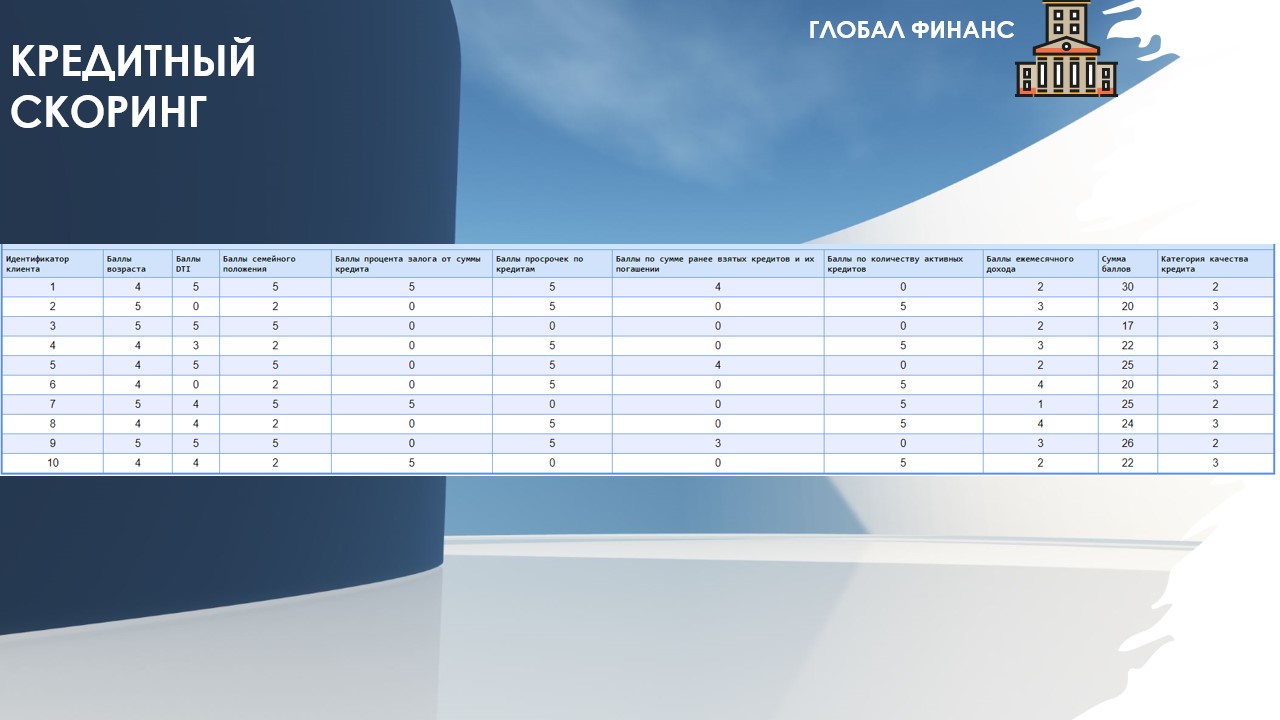
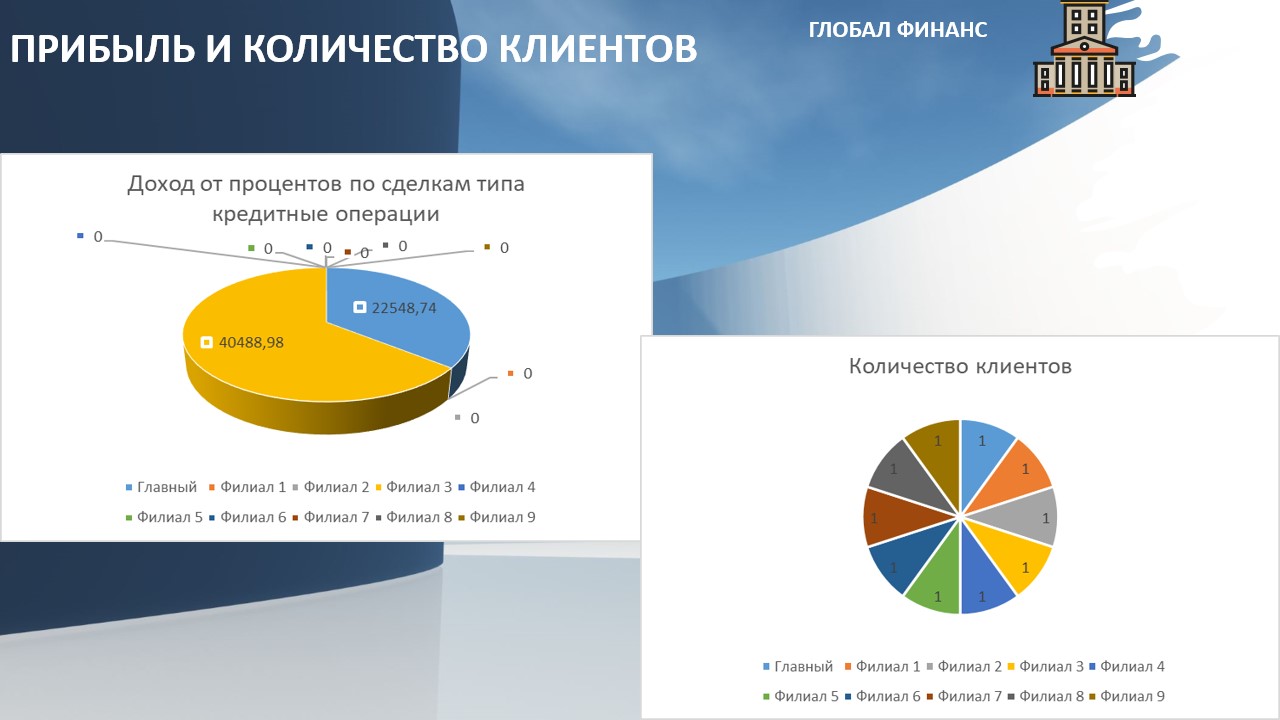
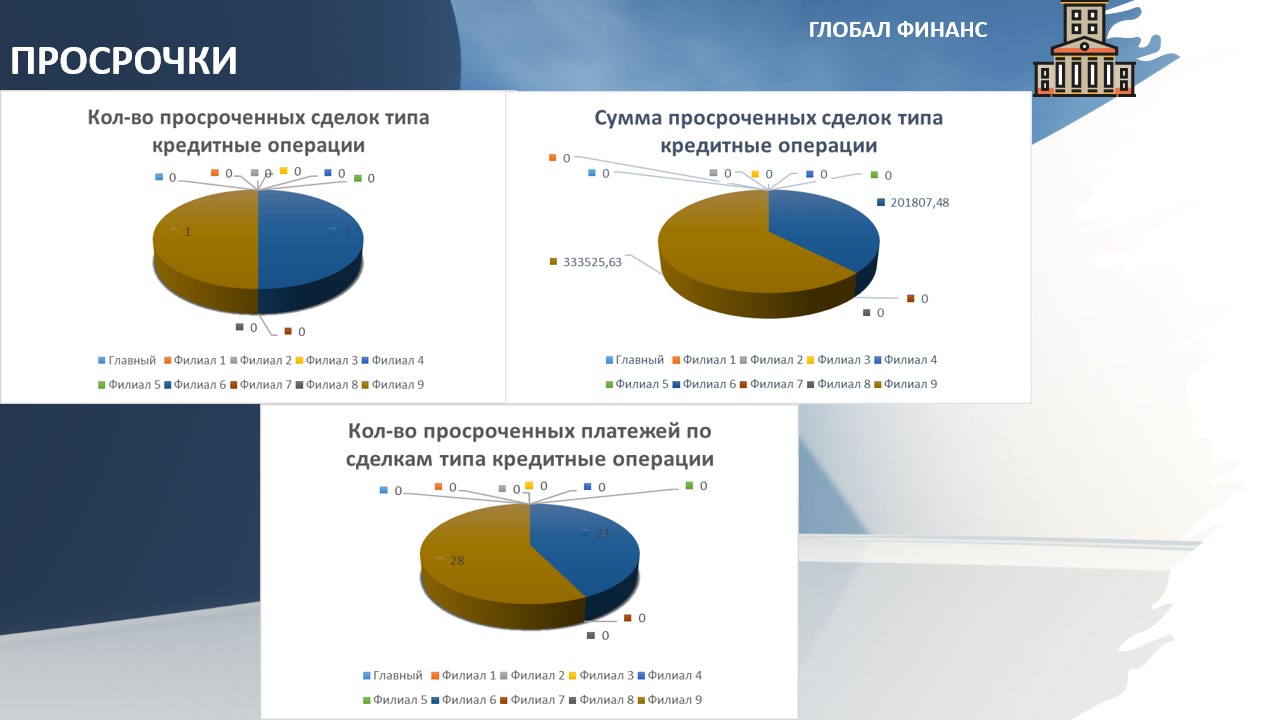
Вторая экранная форма “Оценивание клиента”.



Со стороны клиента экранная форма будет выглядеть так.



Витрины данных вместе с дальнейшей отчетностью:



### 5.1.2. Контуры или наброски\*

не выявлены

## 

## 5.2. Интерфейсы ПО\*

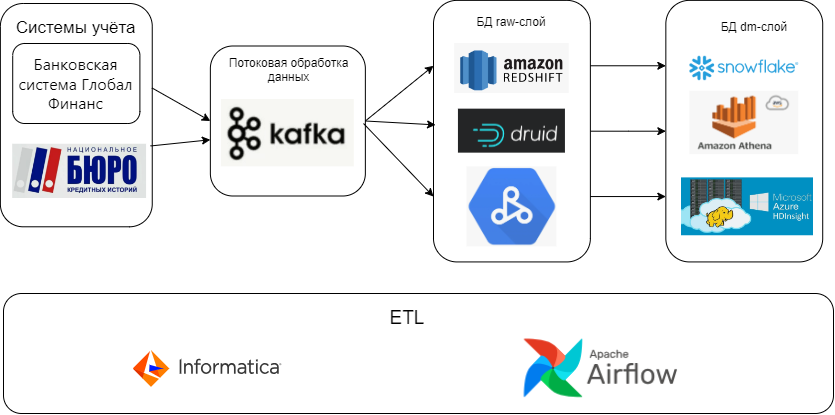
не выявлены

## 5.3. Интерфейсы оборудования (отсутствуют)

## 5.4. Коммуникационные интерфейсы (отсутствуют)

## 5.5. Управление конфигурациями (отсутствуют)

# 6. Платформа и фреймворки



Системы учёта:

Банковская система Глобал Финанс и Бюро кредитных историй (НБКИ) — источники данных, которые передают информацию в потоковую систему для дальнейшей обработки.

Потоковая обработка данных:

Kafka — это распределённая потоковая система, которая позволяет обрабатывать большие объёмы данных в реальном времени. Она может использоваться для различных задач, таких как сбор и анализ логов, обмен сообщениями между системами, потоковая обработка данных и други

Базы данных raw-слоя (слой "сырых" данных):

Amazon Redshift — облачное хранилище данных (Data Warehouse) от Amazon, которое позволяет хранить и анализировать большие объёмы данных.

Druid — распределённая база данных, ориентированная на хранение и обработку временных рядов и аналитических данных в реальном времени.

Apache Kafka — распределённая стриминговая платформа, которая позволяет надёжно передавать потоки данных между системами, а также хранить и обрабатывать их в реальном времени.

Базы данных dm-слоя (слой данных, подготовленных для анализа):

Snowflake — облачная платформа для хранения и аналитики данных, которая объединяет возможности Data Warehouse и Data Lake.

Amazon Athena — сервис от Amazon, который позволяет выполнять SQL-запросы для анализа данных, хранящихся в Amazon S3, без необходимости развертывания серверов.

Microsoft Azure HDInsight — облачная платформа для больших данных от Microsoft, которая поддерживает различные инструменты и фреймворки, включая Apache Hadoop, Spark, Hive и другие для обработки больших данных.

ETL (Extract, Transform, Load):

Apache Airflow — это платформа для управления рабочими процессами (workflow), которая позволяет планировать, отслеживать и координировать выполнение задач. Она предоставляет инструменты для создания и управления DAG (Directed Acyclic Graph) — графами задач, которые определяют порядок выполнения задач в процессе.

Informatica ETL — это инструмент интеграции данных, основанный на архитектуре ETL (Extract, Transform и Load).Он предоставляет программное обеспечение для интеграции данных и услуги для различных предприятий, отраслей промышленности и государственных организаций

Вся эта архитектура представляет собой систему для обработки и хранения данных с использованием стриминговых технологий и облачных решений, которая поддерживает обработку больших данных и выполнение аналитических запросов.

# 7. Тестирование и проверка

Командой были проведены представленные тестирования:

1. Функциональные тесты — проверка основных функций системы на соответствие спецификации.
2. Тестирование производительности — оценка скорости работы системы при различных нагрузках.
3. Нагрузочное тестирование — проверка способности системы обрабатывать большое количество запросов одновременно.
4. Стресс-тестирование — определение пределов возможностей системы и её устойчивости к экстремальным нагрузкам.
5. Юзабилити-тестирование — оценка удобства использования системы пользователями.
6. Тестирование безопасности — проверка защиты системы от несанкционированного доступа и других угроз.
7. Тестирование совместимости — проверка работы системы с различными устройствами и операционными системами.
8. Тестирование интеграции — проверка взаимодействия системы с другими программными продуктами.
9. Регрессионное тестирование — повторное тестирование системы после внесения изменений для проверки того, что новые функции работают корректно и не влияют на уже существующие.

# 8. Критерий приёмки работ

|  |  |
| --- | --- |
| Соответствие результатов техническому заданию и требованиям заказчика | результаты работы должны полностью соответствовать исходным требованиям, чтобы заказчик мог их принять |
| Отсутствие ошибок и дефектов, которые могут повлиять на функциональность системы или её безопасность | Система должна быть безопасной в использование |
| Соблюдение сроков | Все работы должны быть сданы в поставленные дедлайны |
| Проведение тестирование | Весь функционал системы должен быть протестирован . |
| Устранение всех выявленных недостатков и проблем перед сдачей работы заказчику | Если будут выявлены проблемы при сдаче ПСИ их надо исправить. |

# 9. Установка

* Подготовка к установке:
  + Проверить соответствие технических характеристик оборудования требованиям информационной системы.
  + Создать резервную копию данных, которые будут использоваться в системе.
  + Подготовить необходимые для установки инструменты и материалы (например, кабели, сетевые адаптеры).
* Установка программного обеспечения:
  + Загрузить дистрибутив программного обеспечения с официального сайта производителя или из надёжного источника.
  + Запустить процесс установки, следуя инструкциям на экране.
  + Ввести необходимые данные (имя пользователя, пароль, настройки сети и т. д.) в процессе установки.
* Настройка параметров системы:
  + Настроить параметры безопасности (пароль администратора, права доступа пользователей и т. п.).
  + Установить необходимые драйверы и компоненты системы (если требуется).
  + Откорректировать параметры сети (IP-адрес, маска подсети, шлюз по умолчанию и др.).
* Подключение оборудования:
  + При необходимости подключить дополнительные устройства (принтеры, сканеры и т.п.).
  + Убедиться, что оборудование подключено правильно и работает корректно.
* Тестирование системы:
  + Провести тестирование основных функций системы, чтобы убедиться в её работоспособности.
  + Исправить обнаруженные проблемы и повторить тестирование.
* Обучение пользователей:
  + Обучить сотрудников работе с новой системой.
  + Предоставить необходимую документацию и ресурсы для поддержки пользователей.
* Ввод системы в эксплуатацию:
  + Перевести систему в режим эксплуатации.
  + Обеспечить непрерывное функционирование системы и техническую поддержку пользователей.
* Мониторинг и обслуживание:
  + Регулярно проводить мониторинг работы системы для выявления и устранения возможных проблем.
  + Выполнять плановое обслуживание системы (обновление программного обеспечения, резервное копирование данных и т.д.).

## 9.1. Настройка

Настройка системы включает в себя несколько ключевых этапов:

1. Анализ требований: определение целей и задач настройки, а также критериев оценки её успешности.

2. Выбор параметров: выбор конкретных параметров, которые будут изменены в процессе настройки. Это могут быть параметры безопасности, производительности, совместимости и т. д.

3. Изменение параметров: использование инструментов настройки для изменения выбранных параметров. В зависимости от типа системы, это может быть сделано через графический интерфейс, командную строку или другие средства.

4. Тестирование: проверка работоспособности системы после внесения изменений. Тестирование может включать в себя проверку функциональности, безопасности, производительности и других аспектов работы системы.

5. Сохранение настроек: сохранение новых значений параметров в системе. Это позволит сохранить настройки даже после перезагрузки или выключения системы.

## 9.2. Инсталляция

Перед инсталляцией система проходит ПСИ на мощностях команды. После принятия системы заказчиком, система устанавливается на техническом оборудовании заказчика.

# 10. Атрибуты

После окончания разработки система будет обладать такими атрибутами как: портативность , безопасность , расширяемость , интернациональность , поддержка и техническое обслуживание , документация , надежность , доступность , удобство использования

## 10.1. Портативность

Система будет распространяться как на последние версии Windows ,так и на популярные дистрибутивы Linux( Astra Linux , Ubuntu , Arch Linux , Debian , Manjaro)

## 10.2. Производительность

Система будет обеспечивать высокую производительность, с быстрой обработкой запросов и минимальным временем отклика для пользователей.

Она сможет эффективно управлять большим объемом данных (до 100ТБ) и поддерживать высокую скорость выполнения аналитических операций и отчетов.

Система справится с больше сотней одновременно работающих в ней сотрудников и запросов без значительного снижения скорости работы или качества обслуживания.

## 10.3. Безопасность

Система обеспечит защиту данных с помощью шифрования при хранении и передаче информации, а также через надежную аутентификацию и авторизацию пользователей.

Она будет регулярно обновляться для устранения уязвимостей и включать защиту от внешних атак и вредоносных действий.

Реализуется постоянное резервное копирование и восстановление данных, а также вести журналы аудита для отслеживания инцидентов безопасности.

## 10.4. Расширяемость

Система должна быть легко масштабируемой как горизонтально, так и вертикально, чтобы поддерживать рост объема данных и числа пользователей без потери производительности.

Она должна позволять добавление новых функциональных модулей и интеграцию с другими системами по мере необходимости.

Система должна поддерживать гибкую настройку ресурсов, таких как вычислительная мощность и объем памяти, для эффективного реагирования на изменяющиеся нагрузки и требования.

## 10.5. Интернационализация

Система будет поддерживать актуальные технологии использующиеся на территории стран в которых она разворачивается.

Система должна быть локализована под ту страну где она разворачивается.

## 10.6. Поддержка и техническое обслуживание

Команда разработки обязуется поддерживать стабильное состояние системы , обновление и исправление ошибок бесплатно в течении года с выпуска системы

Система должна работать автономно и быстро на объеме данных не превышающий 100ТБ.

## 10.7. Справка и документация

Для сотрудника банка:

1 Этап авторизации: сотрудник банка должен должен зайти в заранее созданный аккаунт в системе

2 Этап проверки формы: в приложении появится заполненная форма которая должна включать в себя ФИО , адрес проживания , семейное положение , DTI , имущество , справку о предыдущих кредитах , ежемесячный доход клиента , на основании этих данных система выдает риск данного кредита , рекомендуемую сумму кредита , сотрудник должен только принять решение о выдаче

3 Дополнительные параметры:

Кнопка архив показывает полную кредитную историю клиента в разных банках включая наш

Кнопка статистика показывает общую статистику данного банка

Для клиента:

После перехода на страницу у клиента будет форма которую надо заполнить параметрами ФИО ,семейное положение ,адрес проживания ,справка о ежемесячном доходе, также клиент может указать имущество в качестве залога и предоставить документы на это имущество , поле комментарий используется для указания дополнительной информации и не обрабатывается системой

После заполнения и отправки формы данные шифруются и отправляются в ближайший филиал банка , где обрабатываются системой , после чего попадают в приложение сотрудников банка

## 10.8. Надежность

Система должна обеспечивать высокую степень надежности, с минимальным числом сбоев и ошибок в работе , время устранения которых не будет превышать 2-х рабочих часов

Она должна иметь встроенные механизмы для регулярного мониторинга и диагностики проблем, а также для автоматического исправления обнаруженных неисправностей.

Система должна поддерживать надежное резервное копирование данных и процедуры восстановления, чтобы гарантировать целостность и доступность информации в случае сбоев или потерь данных.

Она должна демонстрировать устойчивость к внешним и внутренним угрозам, включая защиту от несанкционированного доступа и защиту данных от потерь или повреждений.

Система должна регулярно обновляться и поддерживаться, чтобы соответствовать последним требованиям безопасности и технологическим стандартам, обеспечивая стабильную работу в долгосрочной перспективе.

## 10.9. Доступность

Система должна быть доступна круглосуточно и бесперебойно, с высокими показателями отказоустойчивости и автоматическим восстановлением после сбоев.

Она должна быть совместима с основными операционными системами и браузерами, поддерживать гибкое управление доступом и быть масштабируемой для роста объема данных и числа пользователей.

Система должна соблюдать стандарты безопасности, включая шифрование данных, и предоставлять документацию и поддержку для пользователей.

## 10.10. Требования к удобству использования

Система должна быть простой и логичной в использовании, с понятным размещением элементов управления и навигации, чтобы пользователи могли быстро ориентироваться и эффективно использовать функционал.

Система должна обеспечивать возможность настройки дашбордов и отчетов под индивидуальные потребности пользователей, включая настройку виджетов и выбор отображаемых данных для сосредоточения на наиболее важной информации.

Система должна реализовывать функции импорта и экспорта данных, а также предоставлять предустановленные шаблоны для ввода информации, что сократит количество ошибок и упростит работу с данными.

Система должна включать аналитические инструменты, такие как графики и диаграммы, а также интерактивные отчеты, чтобы облегчить анализ данных и принятие решений, делая информацию более наглядной и понятной.

Система должна быть доступна с мобильных устройств и поддерживать несколько языков, чтобы обеспечить удобство работы в любых условиях и для пользователей из разных стран.

Система должна интегрировать контекстные подсказки и обучающие материалы прямо в интерфейс, а также обеспечивать доступ к технической поддержке для оперативного решения возникающих проблем.