ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ РЕСПУБЛИКИ ХАКАСИЯ

«ХАКАССКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

**ОТЧЕТ**

по учебной практике

по профессиональному модулю

**ПМ.02 ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ**

специальности 09.02.07 Информационные системы и программирование   
Квалификация Технический писатель

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент гр. | ИС(ТП)-31 |  |  |  | Соломатин А.Д. |
|  |  |  | *подпись* |  | *Фамилия И.О.* |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Руководитель практики  от ГБПОУ РХ ХПК |  |  |  |  |  |  | Черкашин Д.С |
|  | *оценка* |  | *дата* |  | *подпись* |  | *Фамилия И.О.* |

Абакан 2025 г

СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc198208438)

[Введение 3](#_Toc198208439)

[1 Анализ предметной области 4](#_Toc198208440)

[1.1 Основные сущности предметной области 4](#_Toc198208441)

[1.2 Основные бизнес-процессы 4](#_Toc198208442)

[1.3 Существующие проблемы и требования 5](#_Toc198208443)

[1.4 Изучение аналогов 5](#_Toc198208444)

[2 Построение функциональной архитектуры программного средства 8](#_Toc198208445)

[2.1 Описание нотации 8](#_Toc198208446)

[2.2 Нотация ER-диаграммы (с «воронкой» Crow’s Foot) 8](#_Toc198208447)

[2.3 Нотация UML Use Case (диаграмма вариантов использования) 8](#_Toc198208448)

[2.4 Описание выбранного CASE-средства 9](#_Toc198208449)

[2.5 Построение ER-диаграммы 10](#_Toc198208450)

[2.6 Построение диаграммы Вариантов использования и диаграммы Последовательности 12](#_Toc198208451)

[2.7 Построение диаграммы Деятельности, диаграммы Классов 14](#_Toc198208452)

[2.8 Построение диаграмм потоков данных 16](#_Toc198208453)

[3 Разработка прототипа 24](#_Toc198208454)

[3.1 Разработка макета интерфейса в figma 24](#_Toc198208455)

[3.2 Изучение работы в системе контроля версий и её настройка 24](#_Toc198208456)

[3.3 Разработка тестового сценария 25](#_Toc198208457)

[3.4 Разработка тестовых пакетов 28](#_Toc198208458)

[Заключение 29](#_Toc198208459)

[Список использованных источников 30](#_Toc198208460)

[Глоссарий 31](#_Toc198208461)

[Список аббревиатур 32](#_Toc198208462)

[ПИЛОЖЕНИЕ А 33](#_Toc198208463)

Введение

В эпоху цифровой трансформации финансовый сектор стремительно внедряет высокотехнологичные решения для оптимизации внутренних процессов и повышения качества обслуживания клиентов. ПАО «Сбербанк» — ведущий универсальный банк Российской Федерации, ежедневно обрабатывающий сотни тысяч операций и обращений со стороны физических и юридических лиц. Автоматизация ключевых бизнес-процессов, в том числе приема и рассмотрения кредитных заявок, становится одним из приоритетных направлений развития банка, направленным на ускорение обработки данных, снижение операционных рисков и улучшение клиентского опыта.

Актуальность работы обусловлена необходимостью создания информационной системы, способной объединить в едином цифровом пространстве сбор, верификацию и анализ кредитных заявок, обеспечивая прозрачность и оперативность принятия решений. Современные клиенты ожидают мгновенного получения ответа на свои запросы, а регуляторные требования Центрального банка Российской Федерации предъявляют жесткие нормы по сохранности и целостности персональных данных. Разработка надежного и масштабируемого решения позволит ПАО «Сбербанк» не только повысить эффективность внутренних операций, но и сохранить лидерские позиции на рынке финансовых услуг.

Целью данной учебно-практической работы является проектирование и описание архитектуры информационной системы автоматизации обработки кредитных заявок в ПАО «Сбербанк». В рамках этой цели предстоит выполнить комплекс мероприятий от формулировки технического задания до создания формальных моделей данных и сценариев использования системы.

Для достижения поставленной цели необходимо решить следующие задачи:

* Провести анализ предметной области, определить основные сущности и их взаимосвязи.
* Разработать техническое задание, включающее описание функций, требований к надежности и сопровождению.
* Построить ER-диаграмму и описать её элементы и связи.
* Оформить диаграмму вариантов использования и таблицу прецедентов.
* Подготовить описание архитектуры решения и предложить рекомендации по дальнейшему развитию.

1 Анализ предметной области

Сбербанк обрабатывает десятки тысяч заявок на кредит ежедневно, взаимодействуя с клиентами по различным каналам: онлайн-банкинг, мобильное приложение и отделения. Каждый клиент предоставляет персональные и финансовые данные, которые необходимо проверить на соответствие внутренним и внешним требованиям (кредитная история, платёжеспособность, наличие обеспечений). В ходе обработки заявки задействованы несколько подразделений: служба безопасности, кредитный комитет и операционные кассы. Задача системы — автоматизировать сбор, верификацию и принятие решений, сократив время обработки и минимизировав ошибки, свойственные ручным процедурам.

1.1 Основные сущности предметной области

Клиент — физическое или юридическое лицо, подающее заявку на кредит; в его описании содержатся персональные и контактные данные. Заявка представляет собой электронный документ с параметрами кредита (сумма, срок, цель), а также со статусом и датой подачи. Кредитный продукт — это предложение банка (ипотека, автокредит, потребительский кредит) с конкретными условиями: процентной ставкой, максимальным лимитом и сроками. Сотрудник Сбербанка (менеджер, аналитик или специалист службы безопасности) участвует в обработке заявки на всех её этапах. Решение отражает итоговый вердикт по заявке (одобрено или отказано) и содержит комментарии и время принятия. История действий представляет собой лог всех операций над заявкой: изменения статуса, комментарии сотрудников и уведомления клиенту.

1.2 Основные бизнес-процессы

* Регистрация и аутентификация клиента  
  Клиент создаёт учётную запись и проходит проверку по базам (паспортные данные, кредитная история).
* Подача заявки на кредит  
  Клиент заполняет форму: выбирает продукт, указывает сумму, срок и цель. Система сохраняет заявку в черновиках или сразу отправляет на обработку.
* Первичная верификация  
  Сотрудник службы безопасности проверяет достоверность введённых данных и отсутствие рисков (протоколы ЧС, санкционные списки).
* Кредитный анализ  
  Аналитик оценивает платёжеспособность: автоматически рассчитывает платёж по формуле аннуитета, проверяет долговую нагрузку.
* Принятие решения  
  Кредитный комитет или уполномоченный сотрудник выносит окончательное заключение и фиксирует его в системе.
* Уведомление клиента  
  Клиент получает PUSH-уведомление, SMS или e-mail с результатом и дальнейшими инструкциями.
* Архивирование и отчётность  
  Все данные сохраняются в хранилище; готовятся отчёты по заявкам за период для руководства и регуляторов.

1.3 Существующие проблемы и требования

* Длительность обработки**:** ручная проверка может занимать от 1 до 5 дней.
* Ошибки при вводе**:** дублирование и неточности из-за ручного заполнения.
* Непрозрачность статусов**:** клиент не всегда сразу получает актуальный статус заявки.
* Разрозненность систем**:** данные хранятся в нескольких независимых базах, что усложняет отчётность.
* Требования регулятора**:** соблюдение стандартов ЦБ РФ по хранению и защите персональных данных.

1.4 Изучение аналогов

При анализе существующих решений автоматизации кредитования были рассмотрены три наиболее известных банковских сервиса.

Tinkoff Credit предлагает полностью безбумажный сервис, доступный как через мобильное приложение, так и через веб‑интерфейс. Большая часть решений по заявкам выносится автоматически всего за 2–3 минуты благодаря встроенным скоринговым алгоритмам, что позволяет клиенту сразу же узнать о вынесенном решении без ожидания оператора. Интуитивно понятный интерфейс облегчает подачу необходимых документов – достаточно сфотографировать паспорт и сделать селфи, после чего система сама персонализирует процентную ставку под профиль заёмщика. Тем не менее в случае сложных или нестандартных запросов живой консультации не предусмотрено, а принадлежность к экосистеме Tinkoff ограничивает возможности тех, кто предпочитает или уже пользуется другими банками. Кроме того, хоть алгоритмы и демонстрируют высокую точность, в редких случаях они могут неправильно оценить благонадёжность клиента и отказать даже лояльному пользователю.

Альфа‑Банк в своём «Кредит онлайн» объединяет мощь веб‑портала и мобильного приложения, проводя автоматическую скоринговую проверку, но при этом дополняя её обязательными KYC‑процедурами с применением электронного паспорта и биометрии. Это обеспечивает высокий уровень безопасности: подписанные электронной подписью документы и скан лица через биометрию практически исключают возможность мошенничества. При необходимости клиент может обратиться за консультацией к менеджеру через чат или видеозвонок, а интеграция с системой «Альфа‑Клик» упрощает оформление платежей и управление действующим займом. Однако более тщательная проверка и сбор документов могут отнять до нескольких часов, а адаптация интерфейса под мобильные экраны в целом уступает специализированным мобильным решениям. Кроме того, жёсткие параметры процентных ставок делают продукт менее гибким в сравнении с полностью автоматизированными конкурентами.

ВТБ реализовал онлайн‑заявку на кредит внутри своего интернет‑банка «ВТБ Онлайн», где доступны различные виды займов. Заявки до 500 000 ₽ обрабатываются полностью автоматически, тогда как при больших суммах часть этапов всё же проходит в ручном режиме: менеджер связывается с клиентом по видеосвязи для уточнения деталей. Такая комбинированная схема позволяет предложить более гибкие условия по суммам и срокам и при этом интегрировать оформление кредита с другими сервисами ВТБ — от инвестиций до страхования. Но ручная обработка крупных займов увеличивает время вынесения решения (от одного до двух дней), а для малого бизнеса и самозанятых ставки часто оказываются менее выгодными, чем у основных конкурентов. Наконец, обширный функционал интернет‑банка делает интерфейс порой громоздким и сложным для быстрого поиска нужного раздела.

2 Построение функциональной архитектуры программного средства

2.1 Описание нотации

В ходе проектирования используются два типа диаграмм: ER-диаграмма для моделирования структуры данных и UML-диаграммы (вариантов использования) для описания поведения системы. Для каждой нотации применяются устоявшиеся графические элементы, позволяющие наглядно и однозначно представить сущности, связи и действия.

2.2 Нотация ER-диаграммы (с «воронкой» Crow’s Foot)

Сущность — это прямоугольник, внутри которого указывается её название (например, «Клиент» или «Заявка»). Атрибуты изображаются либо в виде овалов, присоединённых к сущности, либо строками внутри прямоугольника; первичный ключ помечается звёздочкой (\*), остальные атрибуты записываются обычным текстом. Связь между сущностями проводится в виде линии с «воронками» (crow’s foot) на концах для отображения кардинальности. Символ «||» указывает на обязательное участие (минимум одна запись), «o|» — необязательное участие (от нуля до одной записи), «o{» — произвольное количество элементов (от нуля до многих), а «|{» — как минимум один элемент и более (от одной до многих).

2.3 Нотация UML Use Case (диаграмма вариантов использования)

Актор (Actor) изображается либо как силуэт человечка, либо с помощью ключевого слова actor перед именем (например, actor Клиент) и представляет собой внешний по отношению к системе субъект. Вариант использования (Use Case) отображается овалом с названием прецедента (например, «Подать заявку») и описывает отдельную функциональность системы. Граница системы (System boundary) задаётся большим прямоугольником, окружающим все варианты использования, с подписью, указывающей название системы. Связь между актором и прецедентом (Association) выполняется сплошной линией — обычно без стрелок — что показывает, какие прецеденты доступны данному актору. Между самими прецедентами могут быть отношения двух типов: «include» означает обязательный вызов другого прецедента, а «extend» обозначает условное расширение поведения существующего прецедента.

2.4 Описание выбранного CASE-средства

В качестве CASE-средства для моделирования и документирования архитектурных и поведенческих моделей системы выбрана PlantUML. Это текстовый генератор UML-диаграмм, позволяющий описывать схемы с помощью простого и интуитивно понятного синтаксиса, без необходимости использования тяжеловесных визуальных редакторов.

Преимущества PlantUML:

* Лёгкость интеграции  
  Код диаграмм можно хранить вместе с исходным текстом проекта в системе контроля версий (Git), что обеспечивает прозрачность истории изменений и упрощает совместную работу команды.
* Автоматическая генерация  
  На основе одного текстового файла можно быстро получать ER-диаграммы, диаграммы прецедентов, последовательностей, состояний и другие UML-модели. Обновление диаграммы сводится к редактированию текста.
* Гибкость и расширяемость  
  PlantUML поддерживает множество видов диаграмм, а также интегрируется с CI/CD-конвейерами для автоматической сборки документации.
* Кроссплатформенность  
  Является самостоятельным Java-приложением и работает на Windows, Linux и macOS.

Причины выбора PlantUML для проекта «Автоматизация обработки кредитных заявок»:

* Поддержка версии. Текстовые исходники диаграмм легко отслеживать и возвращаться к предыдущим версиям.
* Читаемость и скорость разработки. Быстрый набор и правка диаграмм с минимальными усилиями, без долгой настройки визуального редактора.
* Универсальность. Возможность описывать как структуру данных (ER-диаграммы), так и сценарии использования (Use Case), диаграммы активности и другие артефакты в рамках одного инструмента.

2.5 Построение ER-диаграммы

Данная ER-диаграмма отражает основные сущности предметной области системы автоматизации обработки кредитных заявок в ПАО «Сбербанк» и связи между ними показаны на рисунке 2.1. Сущность Клиент хранит персональные данные заявителей; один клиент может отправить несколько Заявок. Каждая заявка относится к одному Кредитному\_продукту и может проходить несколько этапов Проверки, выполняемых разными Сотрудниками банка. Кардинальность связей «один-ко-многим» и обязательность участия показаны с помощью «воронки» Crow’s Foot, что позволяет однозначно определить правила целостности данных.

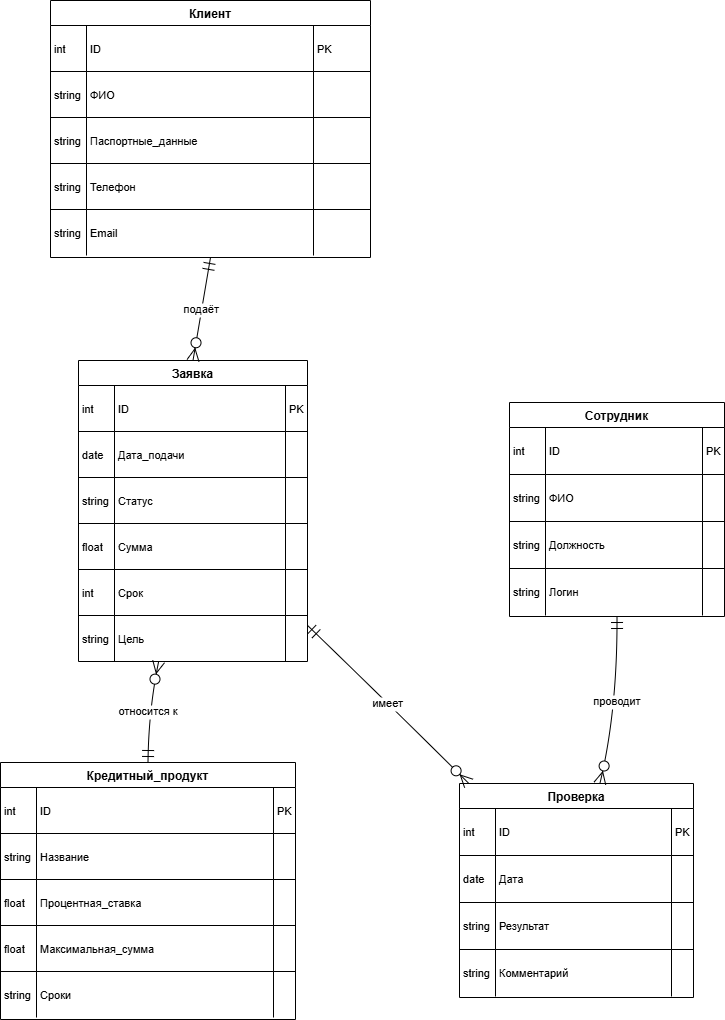


Рисунок 2.1 – ER-диаграмма

В таблицах 1–5 представлены атрибуты основных сущностей предметной области системы автоматизации обработки кредитных заявок. Каждая таблица соответствует одной сущности ER-диаграммы и содержит список полей с их типами и назначением.

Таблица 2.1 – Клиент

| Название | Тип | Описание |
| --- | --- | --- |
| ID | int | Первичный ключ |
| ФИО | varchar(255) | Фамилия, имя и отчество клиента |
| Паспортные\_данные | varchar(50) | Серия и номер паспорта |
| Телефон | varchar(20) | Контактный номер телефона |
| Email | varchar(255) | Адрес электронной почты |

Таблица 2.2 – Заявка

| Название | Тип | Описание |
| --- | --- | --- |
| ID | int | Первичный ключ |
| Дата\_подачи | date | Дата и время создания заявки |
| Статус | varchar(50) | Текущий статус заявки (В обработке, Одобрена, Отказана) |
| Сумма | decimal(12,2) | Запрошенная сумма кредита |
| Срок | int | Срок кредитования в месяц |
| Цель | text | Цель использования кредита |
| Клиент\_ID | int | Внешний ключ, ссылается на таблицу Клиент |
| Продукт\_ID | int | Внешний ключ, ссылается на таблицу Кредитный\_продукт |

Таблица 2.3 – Кредитный\_продукт

| Название | Тип | Описание |
| --- | --- | --- |
| ID | int | Первичный ключ |
| Название | varchar(100) | Наименование кредитного продукта |
| Процентная\_ставка | decimal(5,2) | Годовая процентная ставка |
| Максимальная\_сумма | decimal(12,2) | Максимально допустимая сумма |
| Сроки | varchar(50) | Допустимые сроки кредитования |

Таблица 2.4 – Сотрудник

| Название | Тип | Описание |
| --- | --- | --- |
| ID | int | Первичный ключ |
| ФИО | varchar(255) | Фамилия, имя и отчество сотрудника |
| Должность | varchar(100) | Роль в банке (менеджер, аналитик, служба безопасности) |
| Логин | varchar(50) | Уникальный идентификатор для входа в систему |

Таблица 2.5 – Проверка

| Название | Тип | Описание |
| --- | --- | --- |
| ID | int | Первичный ключ |
| Дата | date | Дата и время проведения проверки |
| Результат | varchar(50) | Итог проверки (Успех, Ошибка, Требуются уточнения) |
| Комментарий | text | Дополнительная информация от проверяющего |
| Заявка\_ID | int | Внешний ключ, ссылается на таблицу Заявка |
| Сотрудник\_ID | int | Внешний ключ, ссылается на таблицу Сотрудник |

2.6 Построение диаграммы Вариантов использования и диаграммы Последовательности

На диаграмме вариантов использования которая показана на рисунке 2.2 (Use Case) отображены основные сценарии взаимодействия пользователей с системой автоматизации обработки кредитных заявок ПАО «Сбербанк». Акторы: Клиент (обращается за кредитом через веб‑ или мобильный интерфейс), Сотрудник банка (проверяет и обрабатывает заявку) и Администратор (управляет пользователями и настраивает систему). Клиент может зарегистрироваться, подать и отслеживать статус заявки. Сотрудник выполняет проверку данных, рассчитывает предварительные условия и выносит решение по заявке. Администратор отвечает за управление учётными записями и просмотр статистических отчётов.

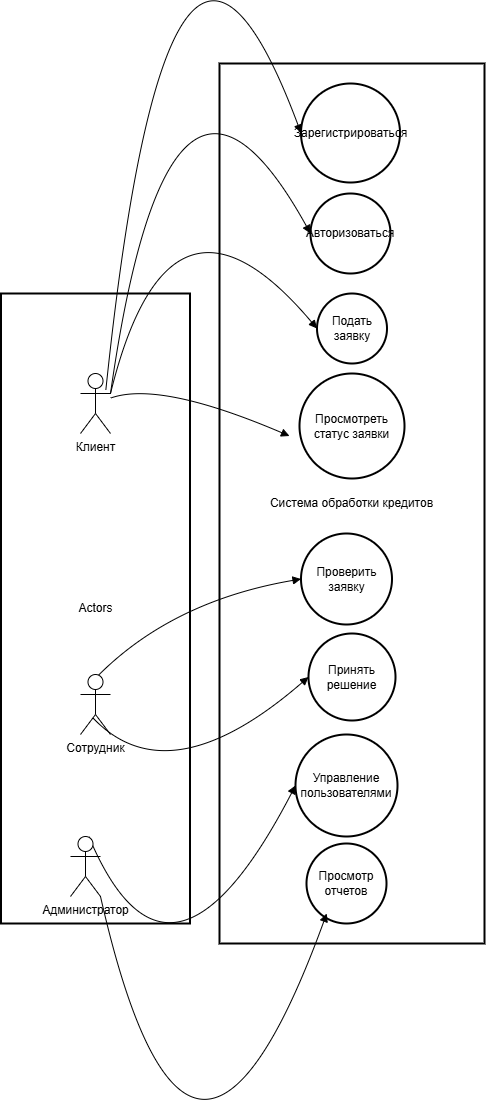


Рисунок 2.2 – Диаграмма вариантов использования системы обработки кредитных заявок ПАО «Сбербанк»

Диаграмма последовательности показана на рисунке 2.3 иллюстрирует процесс подачи кредитной заявки клиентом в системе. Она показывает взаимодействие между основными участниками: клиентом, веб-приложением, базой данных, службой безопасности, кредитным аналитиком и сервисом уведомлений. Сначала клиент заполняет и отправляет форму заявки, после чего веб-приложение сохраняет её в базе данных и инициирует первичную верификацию. По результатам проверки веб-приложение уведомляет кредитного аналитика, а затем с помощью сервиса уведомлений информирует клиента о приёме заявки.

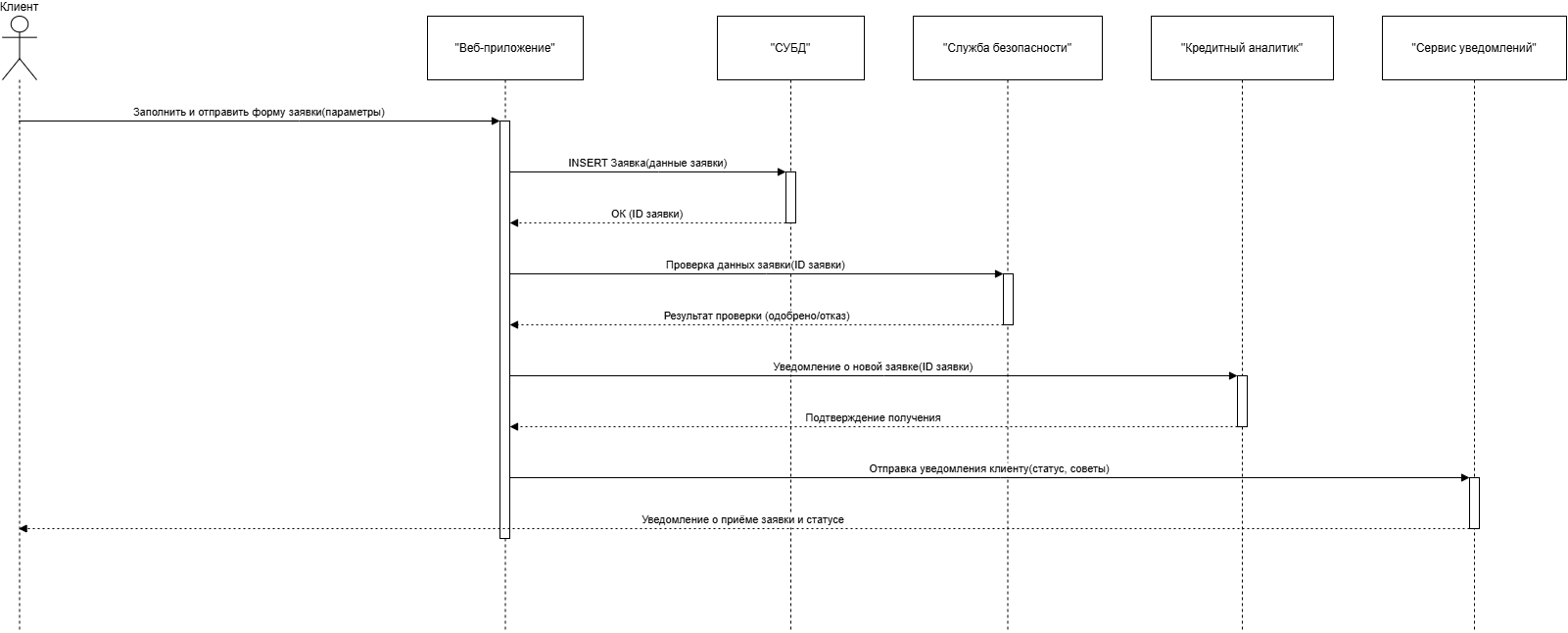


Рисунок 2.3 – Диаграмма последовательности сценария «Подача заявки»

2.7 Построение диаграммы Деятельности, диаграммы Классов

Данная диаграмма деятельности отражает пошаговый алгоритм обработки кредитной заявки в системе. Процесс начинается с подачи заявки клиентом и включает в себя верификацию данных, анализ кредитоспособности, принятие решения и уведомление клиента. Указаны ветвления: при успешной проверке заявка передаётся на анализ, при отказе — сразу уведомляется клиент о результатах. Можно увидеть на рисунке 2.4.

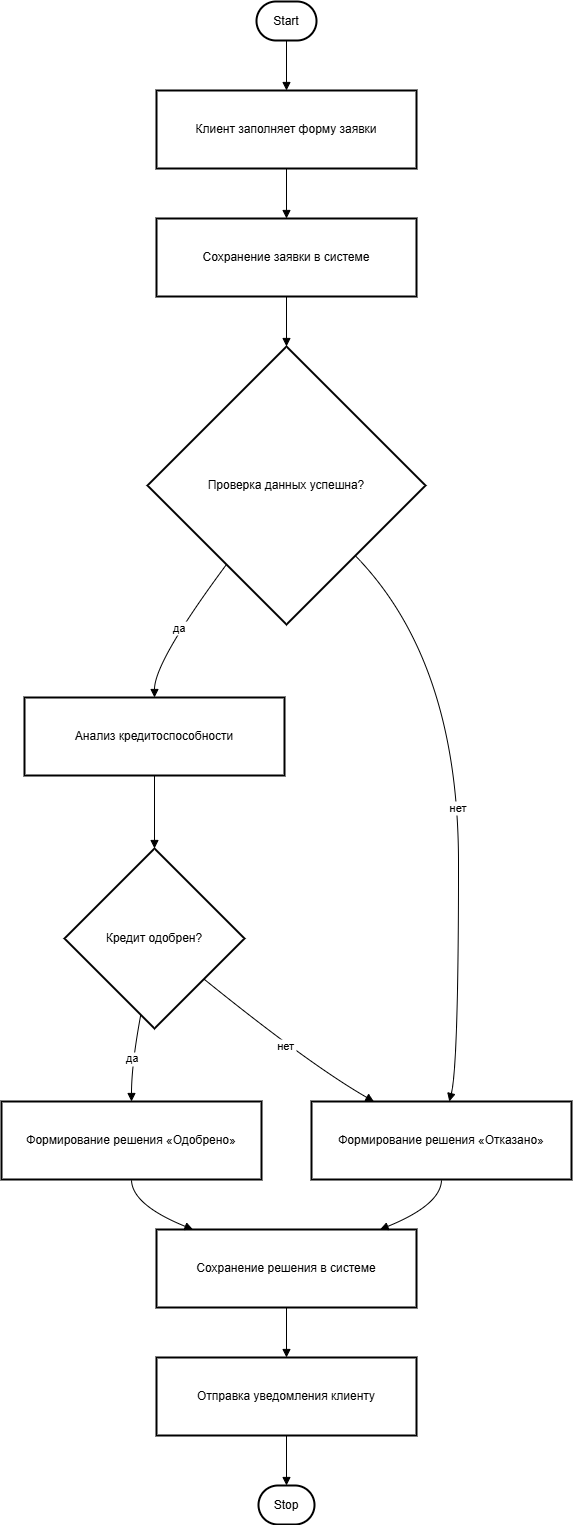


Рисунок 2.4 – Диаграмма деятельности сценария «Обработка заявки»

Класс-диаграмма изображенная на рисунке 2.5 демонстрирует основные статические структуры системы на русском языке и с понятными методами, отражающими предметную область. Классы соответствуют сущностям ER-диаграммы: Клиент, Заявка, Кредитный\_продукт, Сотрудник, Проверка и Сервис\_уведомлений. Методы классов описаны на русском и соответствуют основным операциям: регистрация, подача заявки, верификация, принятие решения и уведомление клиента.

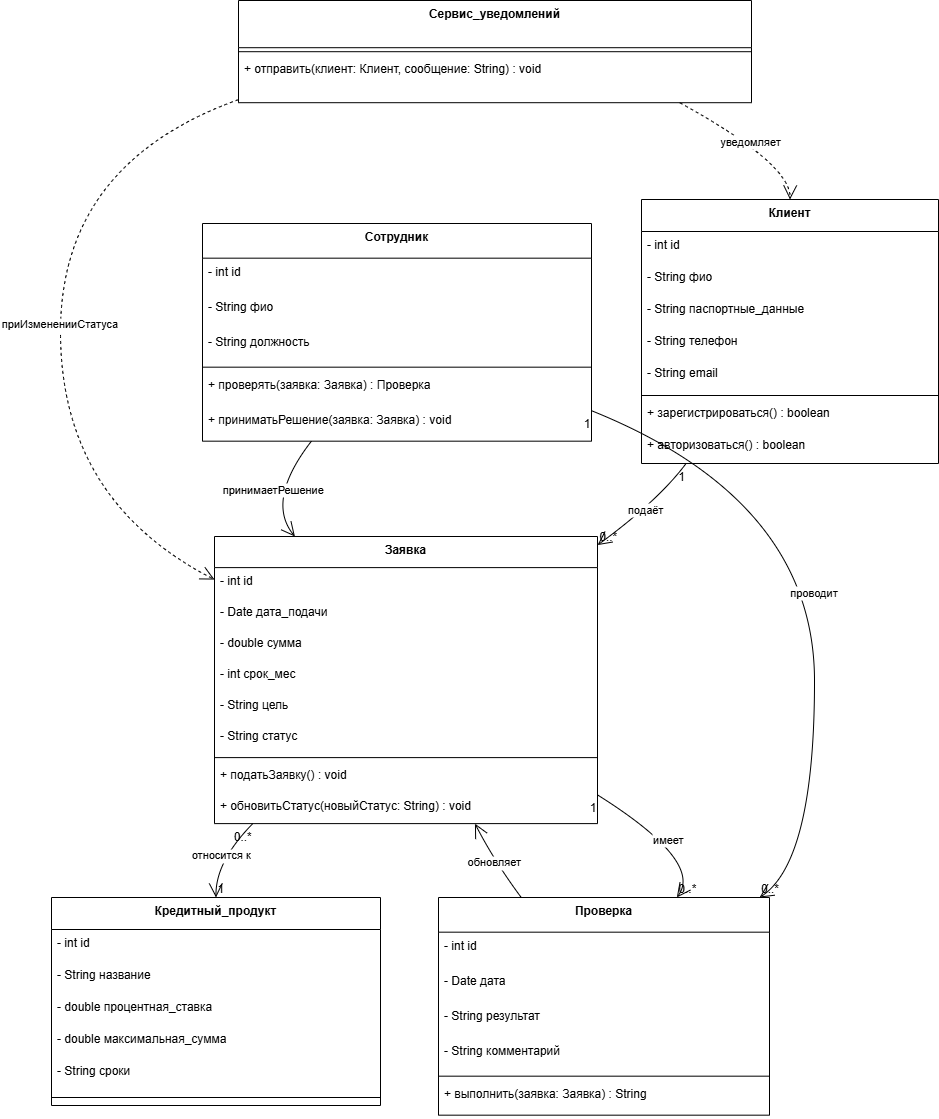


Рисунок 2.5 – Диаграмма классов системы обработки кредитных заявок ПАО «Сбербанк»

2.8 Построение диаграмм потоков данных

Диаграммы потоков данных DFD наглядно отображают на рисунке 2.6, как информация перемещается между внешними сущностями, процессами и хранилищами данных в системе автоматизации обработки кредитных заявок. DFD уровня 0 показывает общую «чёрную коробку» системы: основные потоки от клиента и к базе данных.



Рисунок 2.6 – DFD уровня 0 системы обработки кредитных заявок

DFD уровня 1 раскрывает внутренние процессы: приём и валидацию заявки, анализ кредитоспособности, формирование решения и уведомление клиента. Можно увидеть на рисунке 2.7

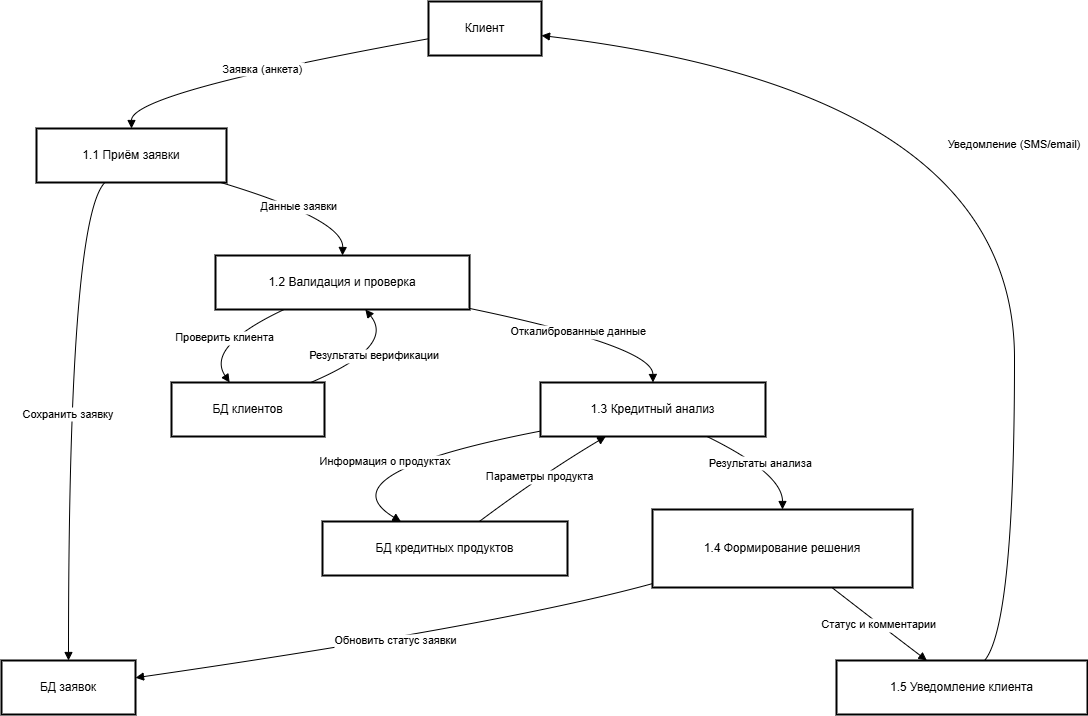


Рисунок 2.7 – DFD уровня 1 процесса «Обработка заявки»

Таблица 2.6 – Прецеденты использования

| № | Название прецедента | Участник | Описание |
| --- | --- | --- | --- |
| 1 | Регистрация пользователя | Клиент | Клиент вводит персональные данные и создаёт учётную запись в системе. |
| 2 | Авторизация в системе | Клиент | Клиент вводит логин и пароль, система проверяет данные и предоставляет доступ. |
| 3 | Подача кредитной заявки | Клиент | Клиент заполняет параметры (сумма, срок, цель) и отправляет заявку на рассмотрение. |
| 4 | Просмотр статуса заявки | Клиент | Клиент запрашивает текущий статус своей заявки и получает информацию из системы. |
| 5 | Просмотр списка заявок | Сотрудник | Сотрудник банка получает список всех заявок, доступных для обработки. |
| 6 | Валидация данных клиента | Сотрудник | Служба безопасности проверяет достоверность паспортных и финансовых данных клиента. |
| 7 | Кредитный анализ заявки | Сотрудник | Аналитик рассчитывает платёж, проверяет кредитную историю и оценивает риски. |
| 8 | Принятие решения по заявке | Сотрудник | Сотрудник или кредитный комитет выносит окончательное решение — одобрить или отказать. |
| 9 | Уведомление клиента о результате | Сервис уведомлений | Система отправляет клиенту SMS или email с информацией о принятом решении. |
| 10 | Управление пользователями | Администратор | Администратор создаёт, изменяет и удаляет учётные записи пользователей и назначает роли. |
| 11 | Настройка параметров продуктов | Администратор | Администратор добавляет или редактирует кредитные продукты: ставки, лимиты, сроки. |
| 12 | Просмотр отчётов | Администратор | Администратор формирует и просматривает отчёты по заявкам и эффективности обработки. |

Ниже приведена таблица основных сценариев взаимодействия «пользователь – система» в модуле подачи и обработки кредитной заявки.

Таблица 2.7 – Сценарии взаимодействия

| № | Действие пользователя | Реакция системы |
| --- | --- | --- |
| 1 | Открыть страницу регистрации | Показать форму регистрации (ФИО, e-mail, пароль) |
| 2 | Ввести данные и нажать «Зарегистрироваться» | Проверить уникальность e-mail, создать учётную запись, показать Личный кабинет |
| 3 | Авторизоваться | Проверить логин/пароль, загрузить профиль клиента |
| 4 | Выбрать тип кредита и заполнить анкету | Валидация полей (сумма, срок), сохранить заявку в статусе «Ожидание» |
| 5 | Нажать «Отправить заявку» | Сохранить заявку в БД, инициировать процесс проверки |
| 6 | Сотрудник открывает список новых заявок | Выгрузить список заявок со статусом «Ожидание» |
| 7 | Сотрудник выбирает заявку и запускает проверку | Выполнить скоринг и KYC-проверку, обновить статус заявки |
| 8 | Сотрудник принимает решение «Одобрить/Отказать» | Зафиксировать решение, оповестить клиента через e-mail и SMS |
| 9 | Клиент проверяет свой статус | Отобразить актуальный статус заявки и дату последнего изменения |

Данная диаграмма показанная на рисунке 2.8 иллюстрирует шаги взаимодействия клиента с системой при создании новой учётной записи. Сначала клиент открывает форму регистрации, затем веб-приложение сохраняет его данные в базе и отправляет письмо для подтверждения e-mail. Последовательность подчёркивает асинхронное взаимодействие с внешним сервисом рассылки и гарантирует, что статус регистрации меняется только после успешного сохранения в СУБД. Это позволяет выявить возможные точки отказа — например, недоступность сервиса e-mail — и проработать обработку ошибок.

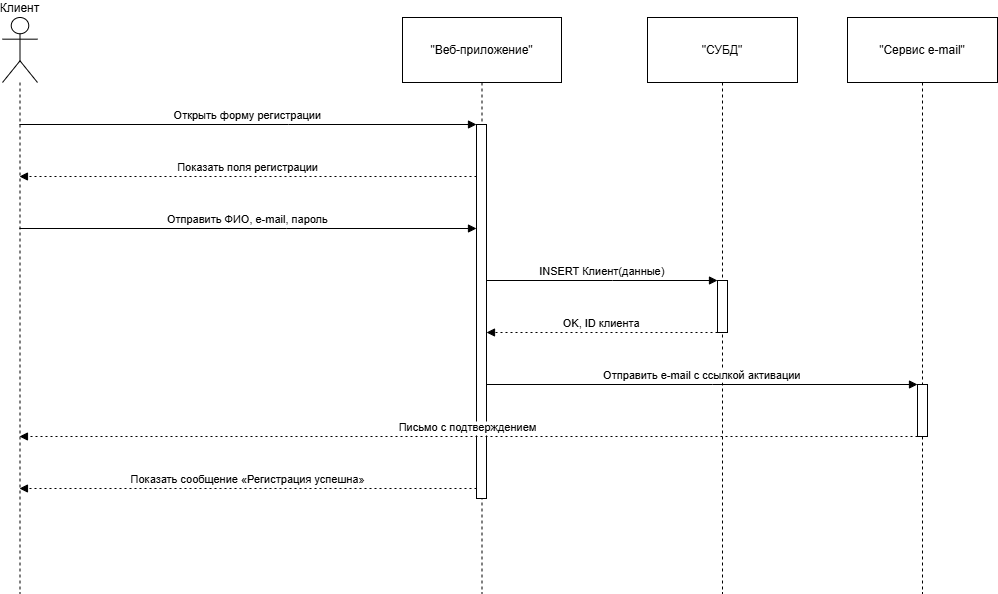


Рисунок 2.8 – Диаграмма «Регистрация клиента»

Диаграмма показанная на рисунке 2.9 демонстрирует процесс создания кредитной заявки: клиент заполняет форму, веб-приложение сохраняет запись и инициирует верификацию. Верификация проходит во внешней службе безопасности, после чего система уведомляет клиента о приёме заявки. Последовательность визуализирует моменты активации каждого участника и потоки данных между ними. Такой обзор помогает понять временные задержки между шагами и спланировать оптимизацию взаимодействий.

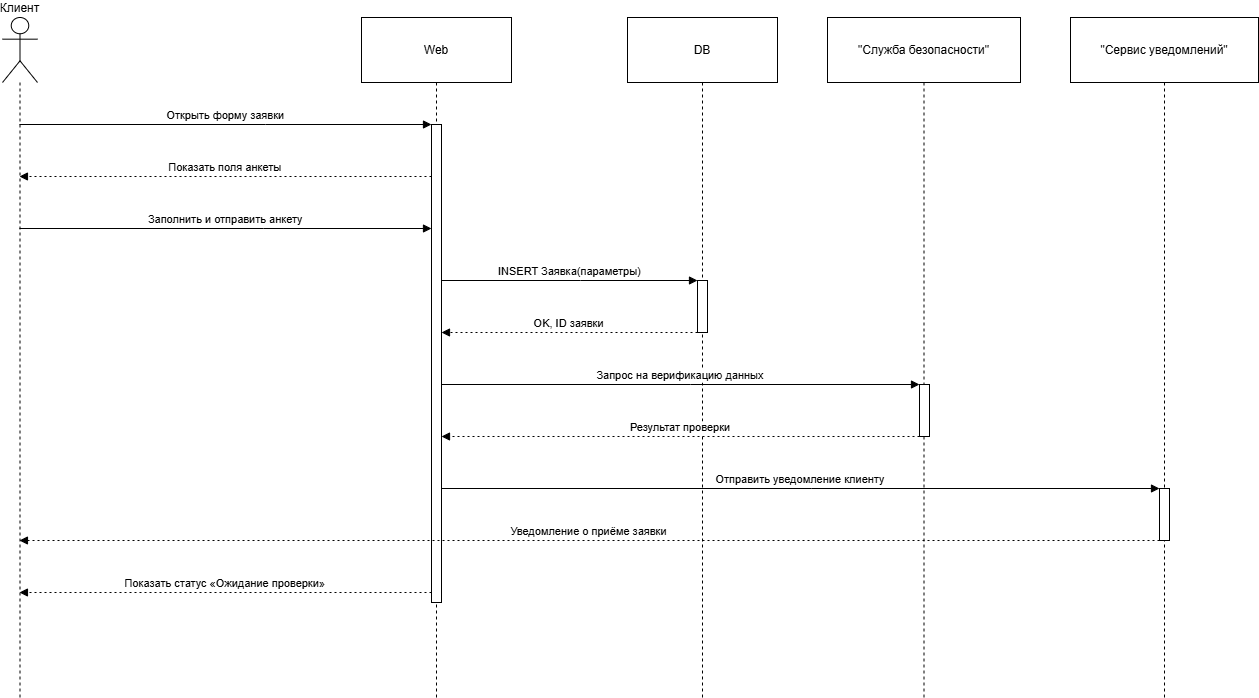


Рисунок 2.9 – Диаграмма «Подача кредитной заявки»

В этой диаграмме на рисунке 2.10 показан сценарий работы банковского сотрудника при окончательном решении по заявке. Сотрудник запрашивает список проверенных заявок, выбирает одну и инициирует анализ платёжеспособности у кредитного аналитика. После получения результатов система сохраняет решение в базе и отправляет клиенту уведомление. Диаграмма подчёркивает зависимость от модуля аналитики и необходимости синхронизации обновления статуса заявки с отправкой сообщений.

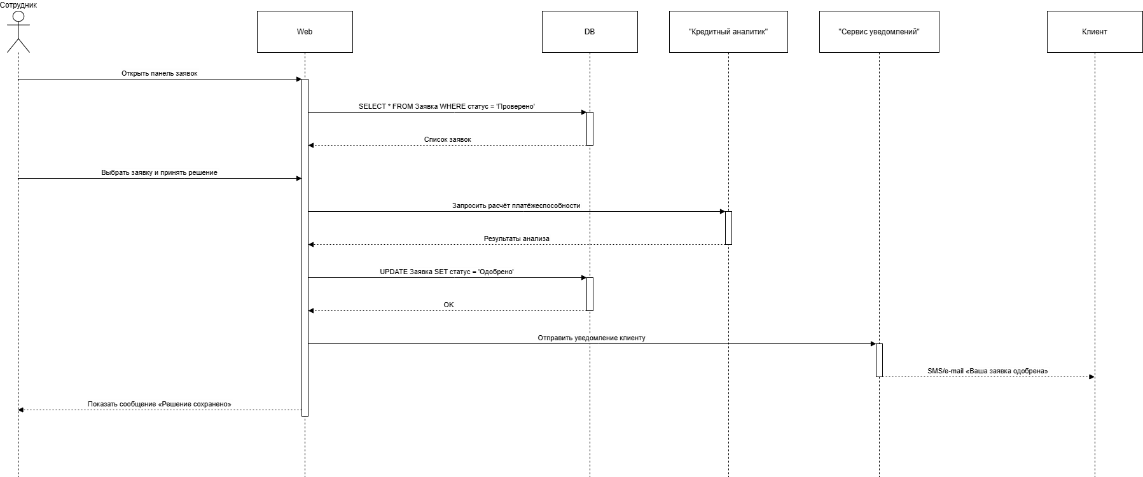


Рисунок 2.10 – Диаграмма «Принятие решения сотрудником»

Диаграмма на рисунке 2.11 активности отображает последовательность операций и ответвлений в процессе обработки кредитной заявки. Она показывает, как система переходит от приёма анкеты к проверке данных, анализу кредитоспособности и формированию решения. Ветвления отражают разные сценарии: успешная верификация, отказ при проверке и длительная ручная обработка при крупных суммах. Эта модель помогает выявить узкие места и оптимизировать логику автоматизации.

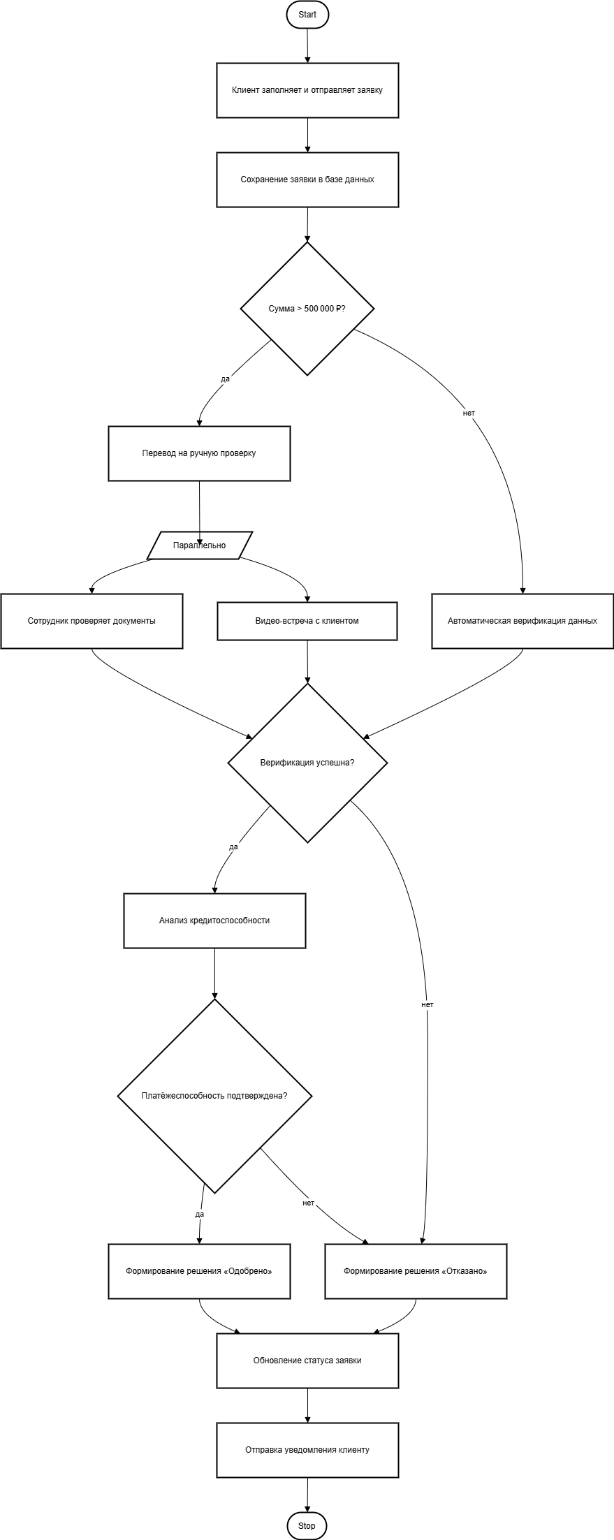


Рисунок 2.11 – Диаграмма активности

Диаграмма состояний отражена на рисунке 2.12 иллюстрирует жизненный цикл объекта «Заявка» и переходы между ключевыми статусами в системе. Каждое состояние отражает этап обработки: от создания и проверки до вынесения решения и архивирования. События и условия переходов (например, успешная верификация, прием решения) документируют логику изменения статуса. Такой подход помогает гарантировать корректную маршрутизацию заявок и упростить поддержку бизнес‑правил.

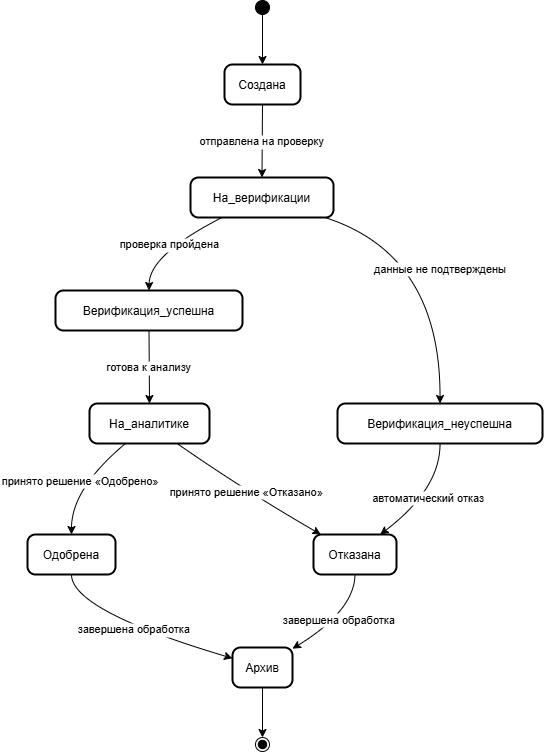


Рисунок 2.12 – Диаграммасостояний

DFD-диаграмма на рисунке 2.13 показывает, как данные перемещаются между внешними участниками (клиентом, банковскими службами) и внутренними процессами системы. Уровень 0 демонстрирует общую инфраструктуру: приём заявки, проверка, решение и хранение в базе. Для углубленного понимания можно детализировать DFD уровня 1, разделяя проверку на автоматические и ручные этапы. Такой подход позволяет оптимизировать потоки данных и выявить узкие места в интеграции с внешними сервисами.

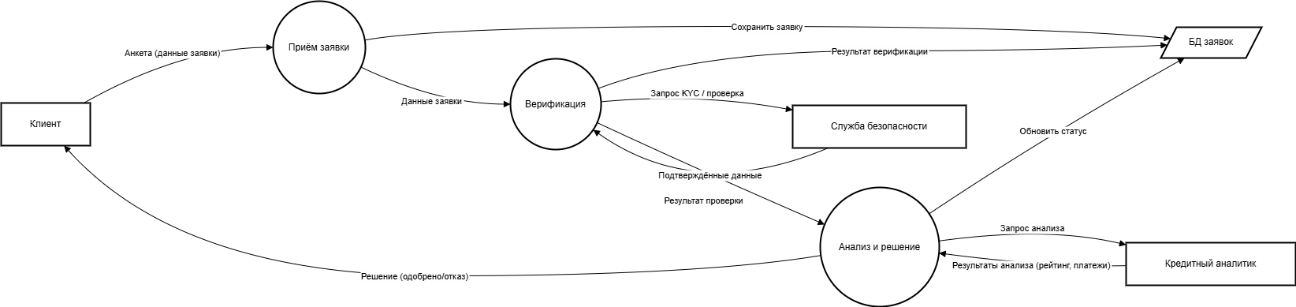


Рисунок 2.13 – DFD-диаграмма

3 Разработка прототипа

3.1 Разработка макета интерфейса в figma

На рисунке 3.1 представлен макет главной страницы веб-сайта автоматизации кредитных заявок, выполненный в формате 16:9. В верхней части экрана находится светлая шапка с логотипом «СберБанк» и навигационным меню, позволяющим быстро перейти к основным разделам. Центральная зона занимает яркий геро-блок с градиентным фоном, крупным заголовком и призывом «Начать сейчас», подчёркивая фокус на мгновенной подаче заявки. Ниже расположены три карточки преимуществ сервиса — «0% комиссия», «Онлайн-решение за 5 минут» и «Поддержка 24/7» — выполненные в модной цветовой палитре и скруглёнными углами. Внизу страницы находится тёмный футер с копирайтом, завершающий современный и молодежный дизайн макета.

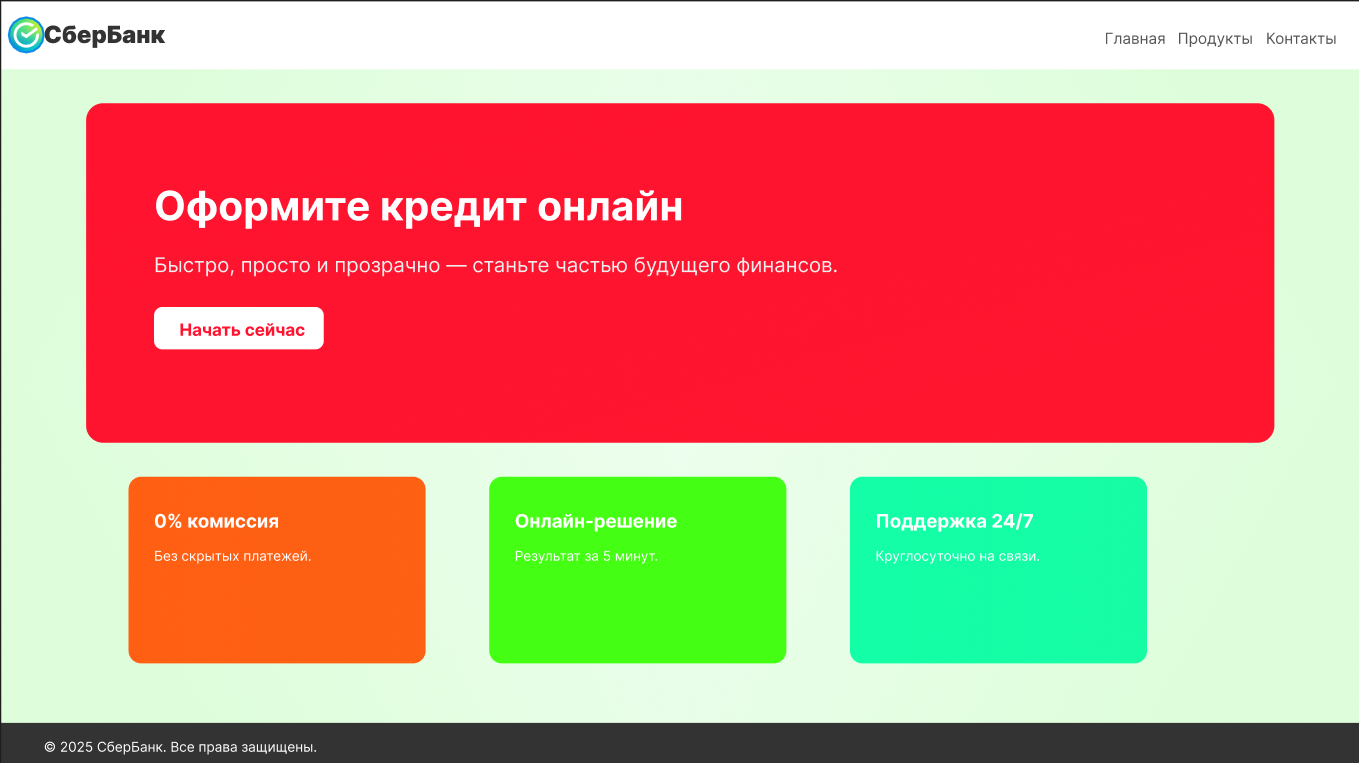


Рисунок 3.1 – Макет интерфейса сайта в Figma

3.2 Изучение работы в системе контроля версий и её настройка

На рисунке 3.2 показан ответ github на вызов что link установлен

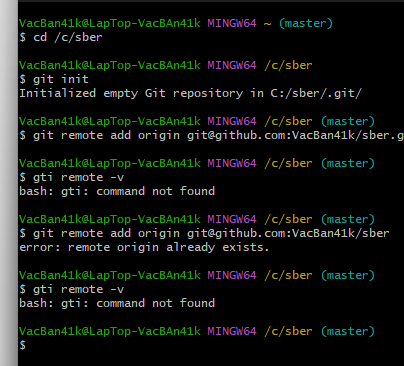


Рисунок 3.2 – GitHub

Для начала работы с Git и GitHub через Git Bash нужно открыть терминал и перейти в рабочую папку вашего проекта командой cd /c/путь/до/вашего/проекта. Это переместит вас в корневой каталог локального проекта. Далее нужно инициализировать репозиторий с помощью команды git init. Эта команда создаст скрытую папку .git и начнёт отслеживать изменения в проекте.

Затем необходимо добавить все файлы проекта в индекс с помощью команды git add ., что подготовит их к первому коммиту. После этого можно сделать первичный коммит с описанием, например: git commit -m "Первоначальный коммит: добавлен весь проект". Это зафиксирует текущее состояние проекта в истории версий.

По умолчанию Git создаёт ветку с именем master, но современные стандарты рекомендуют использовать main, поэтому ветку можно переименовать командой git branch -M main.

На следующем этапе нужно связать локальный репозиторий с удалённым на GitHub. Для этого используется команда git remote add origin git@github.com:VacBan41k/UP.git, где указан SSH-адрес вашего репозитория. После этого вы сможете отправлять изменения на GitHub с помощью команды git push.

3.3 Разработка тестового сценария

Ниже список из 60 ключевых тестовых сценариев (TC1–TC60), описывающих проверку функционала ИС «Автоматизация обработки кредитных заявок».

Таблица 3.1 – Тестовые сценарии

| № | Шаг тестирования | Входные данные | Ожидаемый результат |
| --- | --- | --- | --- |
| 1 | Регистрация клиента с валидными данными | ФИО, email=[test@mail.ru](mailto:test@mail.ru), пароль=Qwerty123 | Успешная регистрация, переход в ЛК |
| 2 | Регистрация клиента с уже существующим email | email=[test@mail.ru](mailto:test@mail.ru) | Ошибка «email уже зарегистрирован» |
| 3 | Регистрация с некорректным форматом email | email=user@mail | Ошибка «некорректный формат email» |
| 4 | Регистрация с коротким паролем (<6 символов) | пароль=12345 | Ошибка «пароль слишком короткий» |
| 5 | Регистрация без обязательных полей | все поля пустые | Ошибка «заполните все обязательные поля» |
| 6 | Авторизация с корректными данными | email=[test@mail.ru](mailto:test@mail.ru), пароль=Qwerty123 | Успешный вход, отображение ЛК |
| 7 | Авторизация с неверным паролем | email=[test@mail.ru](mailto:test@mail.ru), пароль=WrongPass | Ошибка «неверный пароль» |
| 8 | Авторизация незарегистрированного пользователя | email=[noone@mail.ru](mailto:noone@mail.ru) | Ошибка «пользователь не найден» |
| 9 | Восстановление пароля (валидный email) | email=[test@mail.ru](mailto:test@mail.ru) | Письмо со ссылкой для сброса пароля |
| 10 | Восстановление пароля (незарегистрированный email) | email=[fake@mail.ru](mailto:fake@mail.ru) | Ошибка «email не зарегистрирован» |
| 11 | Открытие формы подачи заявки | — | Форма заявки отображается |
| 12 | Подача заявки с валидными параметрами | сумма=100 000, срок=12, цель=«Ремонт» | Статус заявки «В ожидании проверки» |
| 13 | Подача заявки с суммой ниже минимума | сумма=500 | Ошибка «сумма ниже допустимого» |
| 14 | Подача заявки с суммой выше максимума | сумма=15 000 000 | Ошибка «сумма превышает лимит» |
| 15 | Подача заявки с буквой в числовом поле | сумма=«100k» | Ошибка «некорректный формат» |
| 16 | Подача заявки без указания цели | цель пусто | Ошибка «укажите цель кредита» |
| 17 | Сохранение заявки в черновики | заполнены не все поля | Заявка сохранена как черновик |
| 18 | Редактирование черновика заявки | изменить сумму | Черновик обновлен |
| 19 | Удаление черновика | нажать «Удалить черновик» | Черновик удален |
| 20 | Попытка редактирования отправленной заявки | заявка в статусе «В ожидании проверки» | Редактирование недоступно |
| 21 | Просмотр списка заявок клиентом | — | Отображается все заявки клиента |
| 22 | Фильтрация заявок по статусу | выбрать «Одобрено» | Отображаются только одобренные заявки |
| 23 | Сортировка заявок по дате | сортировка по «дате подачи» | Список упорядочен по дате |
| 24 | Сотрудник просматривает ожидающие заявки | — | Показаны заявки со статусом «В ожидании проверки» |
| 25 | Верификация паспорта (валидные данные) | паспортные данные клиента | Статус проверки «успешно» |
| 26 | Верификация паспорта (недостоверные данные) | фальшивый паспорт | Статус проверки «неуспешно» |
| 27 | Логирование результатов проверки | — | Запись в журнал о проверке |
| 28 | Расчет аннуитетного платежа | сумма=100 000, срок=12 | Месячный платёж рассчитан верно |
| 29 | Скоринг при плохой кредитной истории | КИ клиента низкая | Автоматический отказ |
| 30 | Просмотр истории проверок | — | Отображаются все этапы проверки заявки |
| 31 | Аналитик оставляет комментарий | комментарий=«нужно уточнение» | Комментарий сохранён |
| 32 | Принятие решения «Одобрено» | нажать «Одобрить» | Статус заявки обновлён на «Одобрено» |
| 33 | Принятие решения «Отказано» | нажать «Отказать» | Статус заявки обновлён на «Отказано» |
| 34 | Отправка e-mail при смене статуса | статус="Одобрено" | Клиент получает письмо |
| 35 | Отправка SMS при смене статуса | номер=+7 123 456 78 90 | Клиент получает SMS |
| 36 | Ошибка при отправке SMS (некорректный номер) | номер=12345 | Лог «не удалось отправить SMS» |
| 37 | Уведомление в ЛК клиента | смена статуса заявки | На панели уведомлений появляется новое сообщение |
| 38 | Добавление нового сотрудника | ФИО, должность, логин | Сотрудник создан |
| 39 | Редактирование профиля сотрудника | смена должности | Профиль обновлён |
| 40 | Удаление сотрудника | нажать «Удалить» | Сотрудник удалён |
| 41 | Проверка разграничения ролей | роли: клиент/сотрудник/админ | Доступы соответствуют ролям |
| 42 | Переход по несуществующей ссылке | URL=/invalid | Отображается страница 404 |
| 43 | Истечение сессии | 30 мин бездействия | Требуется повторная авторизация |
| 44 | Выход из системы (Logout) | нажать «Выйти» | Пользователь разлогинен |
| 45 | CSRF-защита при отправке форм | POST без CSRF-токена | Ошибка «CSRF token missing» |
| 46 | SQL-инъекция в полях формы | сумма="100; DROP TABLE" | Запрос отклонён, зафиксирован инцидент |
| 47 | Проверка хеширования пароля | сохранён пароль | Пароль хранится в базе в виде хеша |
| 48 | Шифрование персональных данных | паспортные данные | Поле шифруется при сохранении |
| 49 | Ежедневное резервное копирование | — | Бэкап базы данных завершён успешно |
| 50 | Восстановление данных из резервной копии | ID заявки | Заявка успешно восстановлена |
| 51 | Время отклика API | GET /api/applications | Среднее время ответа < 200 мс |
| 52 | Нагрузочное тестирование | 1000 запросов/сек | Система устойчива, нет падений |
| 53 | Отказоустойчивость при падении БД | эмуляция сбоя БД | Поднятие резервного узла, продолжение работы |
| 54 | Логирование всех CRUD-операций | любые действия | Все операции логируются |
| 55 | Переключение языка RU/EN | выбор «EN» | Интерфейс на английском языке |
| 56 | Проверка SEO-метаданных | просмотр кода страницы | Теги <title>, <meta> заполнены |
| 57 | Кроссбраузерное тестирование | Chrome, Firefox, Edge, Safari | Интерфейс корректно отображается во всех браузерах |
| 58 | Адаптивность: мобильный экран 320×480 | эмуляция мобильного устройства | Элементы UI адаптированы |
| 59 | Адаптивность: desktop экран 1920×1080 | изменение размеров окна | Интерфейс корректно масштабируется |
| 60 | Интеграция с внешним скоринг-сервисом | эмулятор скоринга | Результаты обрабатываются и сохраняются корректно |

3.4 Разработка тестовых пакетов

Ниже список из 5 тестовых пакетов, описывающих проверку функционала ИС «Автоматизация обработки кредитных заявок».

Таблица 3.2 – Тестовые пакеты

| № | Название пакета | Включённые тест‑кейсы | Краткое описание |
| --- | --- | --- | --- |
| 1 | Аутентификация и безопасность | TC1–TC10, TC43–TC47 | Проверка регистрации, авторизации, восстановления пароля, защиты форм (CSRF, SQL‑инъекции), хеширования и шифрования данных. |
| 2 | Управление заявками (клиент) | TC11–TC23, TC37 | Подготовка, подача, сохранение черновиков, просмотр, фильтрация и сортировка заявок на стороне клиента. |
| 3 | Обработка заявок (сотрудник) | TC24–TC33 | Верификация данных, скоринг, комментирование, принятие решений («Одобрено»/«Отказано»), логирование этапов проверки. |
| 4 | Уведомления и интеграции | TC34–TC36, TC59–TC60 | Отправка e‑mail и SMS, обработка ошибок внешних сервисов, тестирование API и интеграции со скоринг‑сервисом. |
| 5 | Нефункциональное тестирование и надёжность | TC49–TC58 | Резервное копирование и восстановление, время отклика API, нагрузочное и отказоустойчивое тестирование, кросс‑браузерность, адаптивность, SEO. |

Заключение

В ходе выполнения учебно‑практической работы по профессиональному модулю ПМ.02.01 «Осуществление интеграции программных модулей» для ПАО «Сбербанк» была разработана комплексная информационная система автоматизации обработки кредитных заявок. В рамках работы выполнены следующие ключевые этапы:

* Формулирование технического задания  
  Определены термины и основные понятия системы, сформулированы функциональные и нефункциональные требования, требования к дизайну и интерфейсу, надёжности, сопровождению и поддержке.
* Анализ предметной области  
  Выделены и описаны основные сущности (Клиент, Заявка, Кредитный продукт, Сотрудник, Проверка, Сервис уведомлений), основные бизнес‑процессы и проблемы существующей ручной обработки.  
  Проведён сравнительный анализ аналогичных решений конкурентов (Tinkoff Credit, Альфа‑Банк, ВТБ) с указанием их сильных и слабых сторон.
* Проектирование архитектуры и моделей  
  Построены ER‑диаграмма и таблицы атрибутов для формализации структуры данных.  
  Составлены UML‑модели: диаграммы вариантов использования, последовательности, деятельности, состояния и классов.  
  Разработаны DFD‑диаграммы (уровня 0 и 1) для наглядного описания потоков данных и взаимодействия компонентов.
* Разработка прототипа интерфейса  
  Создан макет главной страницы в формате 16:9 с адаптивным дизайном, современными градиентными элементами и чёткой навигацией.  
  Подобрана цветовая схема, отражающая фирменный стиль банка, и продуманы карточки преимуществ сервиса.
* Организация разработки и тестирование  
  Настроена работа в Git/GitLab с моделью Git Flow, реализован CI/CD‑конвейер для автоматического линтинга, тестирования и деплоя.  
  Разработан обширный тест‑план: 60 тест‑кейсов и 5 тематических тестовых пакетов (аутентификация, клиентские операции, обработка сотрудником, интеграции, нефункциональное тестирование).  
  Определены метрики качества: покрытие тестами, время отклика, устойчивость к сбоям, безопасность данных и др.

Цель работы — спроектировать и описать архитектуру ИС автоматизации обработки заявок в Сбербанке — полностью достигнута. Решены все поставленные задачи: от анализа предметной области до разработки моделей, макета интерфейса и комплексного плана тестирования.

Перспективы развития проекта:

* Интеграция с внешними скоринговыми и анти‑фрод сервисами для повышения точности оценки рисков.
* Внедрение машинного обучения и AI‑моделей для предиктивного анализа и динамической настройки условий кредитования.
* Разработка мобильного приложения для клиентов и сотрудников с офлайн‑режимом работы в отделениях.
* Постоянный мониторинг производительности, улучшение отказоустойчивости и расширение модулей статистической отчётности.

Список использованных источников

1. Федеральный закон «О Центральном банке Российской Федерации (Банке России)» от 10.07.2002 № 86‑ФЗ. Официальный интернет‑портал правовой информации [Электронный ресурс]. URL: <http://pravo.gov.ru> (дата обращения: 15.05.2025)
2. Постановление Правления Банка России от 03.11.2017 № 79‑П «О правилах предоставления и оформления кредитов (займов) банками». Официальный сайт Банка России [Электронный ресурс]. URL: <https://cbr.ru> (дата обращения: 15.05.2025)
3. Сбербанк. Кредитные продукты. Раздел «Потребительские кредиты» [Электронный ресурс]. URL: https://www.sberbank.ru/ru/person/kredit/consumer (дата обращения: 15.05.2025)
4. OMG. UML® 2.5.1 Specification. Section 6: Use Case Diagrams [Электронный ресурс]. URL: https://www.omg.org/spec/UML/2.5.1 (дата обращения: 15.05.2025)
5. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. Addison‑Wesley Professional, 2018. Глава 4: Диаграммы классов [Печатный ресурс].
6. PlantUML. PlantUML User Guide. Раздел «ER diagram» [Электронный ресурс]. URL: https://plantuml.com/er-diagram (дата обращения: 15.05.2025)
7. Yakovlev A. «Автоматизация кредитования в банках: тенденции и перспективы». Финансовый журнал «Банковские технологии», № 4(56), 2024. Раздел «Цифровые решения» [Печатный ресурс].

Глоссарий

Клиент — физическое или юридическое лицо, обращающееся в банк за получением кредита; взаимодействует с системой через веб‑ или мобильный интерфейс.

Заявка на кредит — электронный документ, содержащий параметры запрашиваемого кредита (сумма, срок, цель) и персональные данные клиента; проходит этапы верификации, анализа и принятия решения.

Кредитный продукт — тип банковского предложения (потребительский кредит, автокредит, ипотека и пр.) с набором условий: процентная ставка, допустимые суммы и сроки.

Сотрудник — уполномоченное лицо банка (аналитик, менеджер, служба безопасности), выполняющее проверку данных клиента, расчёт платёжеспособности и вынесение окончательного решения.

Проверка (верификация) — процесс подтверждения подлинности и актуальности персональных и финансовых данных клиента; включает KYC‑процедуры и анализ рисков.

Скоринг — автоматизированная оценка кредитного риска клиента на основе статистических моделей и исторических данных (кредитной истории, платёжной дисциплины).

Аннуитет — схема погашения кредита равными периодическими платежами, включающими основную сумму и проценты.

Уведомление — сообщение клиенту о смене статуса заявки, отправляемое через e‑mail, SMS или в виде push‑уведомления в личном кабинете.

API (Application Programming Interface) — интерфейс прикладного программирования, через который модули системы (или внешние приложения) отправляют запросы и получают данные.

ER‑диаграмма (Entity–Relationship) — графическое представление сущностей предметной области и связей между ними; используется для проектирования структуры базы данных.

DFD (Data Flow Diagram) — диаграмма потоков данных, показывающая, как информация перемещается между внешними участниками, процессами и хранилищами данных.

UML (Unified Modeling Language) — унифицированный язык моделирования; включает набор нотаций (Use Case, последовательности, классов, состояний и пр.) для описания структуры и поведения системы.

CI/CD (Continuous Integration / Continuous Deployment) — практика непрерывной интеграции и доставки, при которой изменения в коде автоматически проверяются, тестируются и развёртываются на тестовых или производственных средах.

CRUD‑операции — базовые действия с данными: Create (создание), Read (чтение), Update (обновление), Delete (удаление).

Список аббревиатур

ИС — Информационная система

ПО — Программное обеспечение

ER — Entity–Relationship (сущность–связь)

UML — Unified Modeling Language (унифицированный язык моделирования)

DFD — Data Flow Diagram (диаграмма потоков данных)

API — Application Programming Interface (интерфейс программирования приложений)

CI/CD — Continuous Integration / Continuous Deployment (непрерывная интеграция и непрерывное развертывание)

CRUD — Create, Read, Update, Delete (базовые операции с данными)

KYC — Know Your Customer (процедуры по идентификации клиента)

MTTR — Mean Time to Recovery (среднее время восстановления)

SQL — Structured Query Language (язык структурированных запросов)

HTTPS — Hypertext Transfer Protocol Secure (защищённый протокол передачи гипертекста)

SMS — Short Message Service (служба коротких сообщений)

ID — Identifier (идентификатор)

ПрИЛОЖЕНИЕ А