**РУКОВОДСТВО ПРОГРАММИСТА**

**Бизнес-процесс**

**«Оформление кредитной заявки»**

**Листов 12**

2022

**АННОТАЦИЯ**

В данном программном документе приведено руководство программиста по работе с CRM-системой ELMA365, в которой был разработан бизнес-процесс «Оформление кредитной заявки».

В разделе «Назначение и условия применения программы» указаны назначение и функции, выполняемые программой, условия, необходимые для выполнения программы.

В разделе «Характеристика программы» приведено описание основных

характеристик и особенностей программы (экранные формы, бизнес-процессы).

1. Назначение и условия применения программы

В условиях перехода общества в информационную эпоху, все большую актуальность приобретают автоматизированные системы и бизнес-процессы. Следовательно, целью качественной работы банков является разработка автоматизированной системы работы с клиентами и финансами.

Комплексная автоматизация управления предприятия на сегодняшний день – один из самых эффективных и функциональных инструментов систематизации работы ключевых бизнес-процессов. Зачастую управление бизнес-процессами становится трудоемким, а анализ большого потока первичных данных отнимает много сил, времени.

В условиях современной жизни требуется ускорение процессов обработки информации. Этот процесс подлежит автоматизации, так как обработка информации очень долгий, кропотливый и требующий больших ресурсов процесс.

Обслуживание клиентов, которые хотят оформить кредит, должно проходить быстро, тем самым и регистрация клиента должна оформляться быстро. Следовательно, есть необходимость в ускорении данного процесса.

Главной задачей, которую необходимо было выполнить, было создание бизнес-процесса «Оформление кредитной заявки».

Автоматизация технологического процесса позволила бы осуществлять управление процессом без непосредственного участия человека, либо оставления за человеком права принятия наиболее ответственных решений. Таким образом, решение данной задачи значительно упростило бы работу сотрудников организации и позволило им сконцентрироваться на ключевых моментах процесса.

В ходе производственной практики был спроектирован бизнес-процесс в системе ELMA365, необходимый для автоматизации процесса «Оформление кредитной заявки» на примере небольшого финансового учреждения.

1. Характеристики и работа программы

Требования к логике процесса, которые были выполнены в ходе разработки:

1. Запустить процесс может любой кредитный специалист, указав клиента из справочника с клиентами.
2. Настройка шлюзов по сумме оформления кредитной заявки. Если сумма заявки:

* Меньше 100 тыс. руб. – заявка поступает на согласование начальнику кредитного отдела;
* Меньше 1 млн. руб. – заявка поступает на согласование начальнику кредитного отдела и группе безопасности;
* Свыше 1 млн. руб. – согласование направляется всем вышеперечисленным и начальнику кредитного комитета. Начальник должен организовать совещание кредитного комитета, создав событие и выполнив задачу. По окончанию совещания кредитного комитета должна прийти задача отражения результатов работы комитета. Необходимо реализовать автоматическое создание события встречи кредитного комитета.

1. По окончании согласования – поставить сотруднику отдела продаж задачу уведомления клиента.
2. Передать заявку на выплату к бухгалтеру.
3. Отразить информацию о кредитной истории в карточке клиента.
4. Настройка блока генерации по шаблону (формирование договоров для физических и юридических лиц)
5. Осуществить анализ клиентов с кредитной историей> 1 заявки, вывести их в переменную, которая будет в задаче инициатора.

На рисунке 1 представлена модель бизнес-процесса «Оформление кредитной заявки».

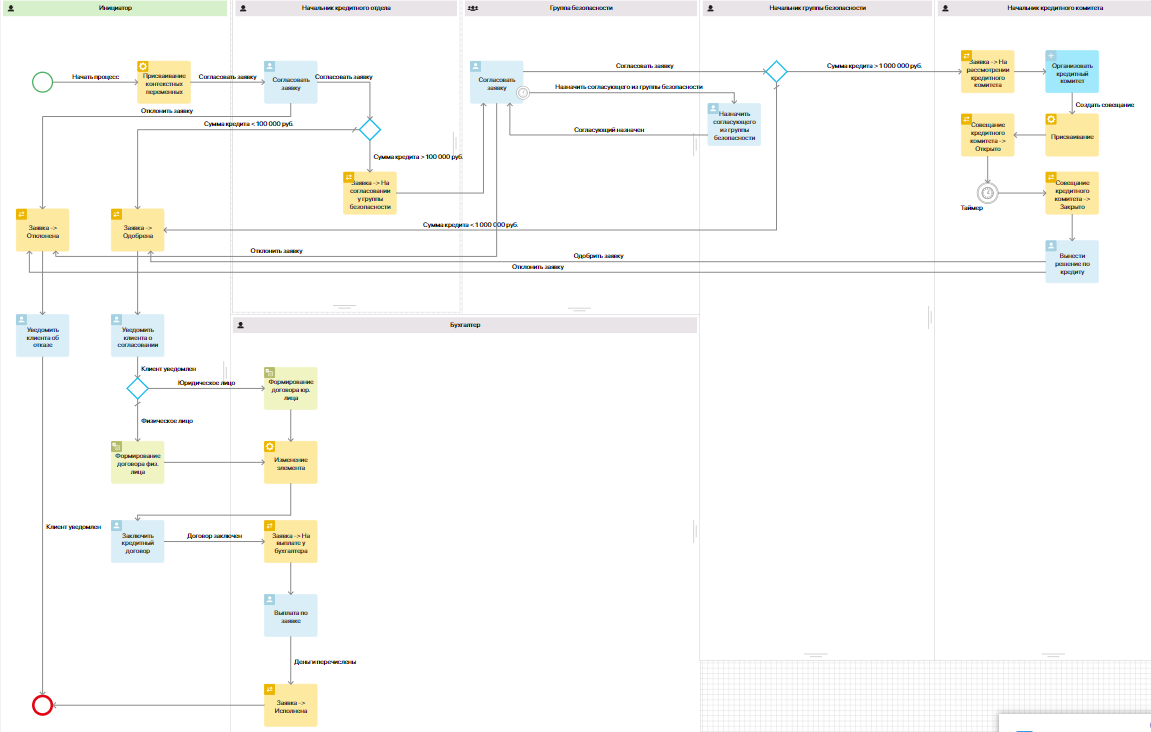


Рисунок 1 – Бизнес-процесс «Оформление кредитной заявки»

В работе данного процесса выделено несколько зон ответственности, это: инициатор, начальник кредитного отдела, группа безопасности, начальник группы безопасности, начальник кредитного комитета и бухгалтер.

Задачи «Согласовать заявку» стандартно разделяются на два варианта развития: «Согласовать заявку» (тогда процесс идёт дальше по сценарию) или «Отклонить заявку» (тогда заявка переходит в статус отклонённой). Также от задачи «Согласовать заявку» может быть разветвление, необходимое для назначения ответственного лица: «Назначить согласующего».

Шлюзы, работающие в зависимости от суммы кредита, также разделяются на два варианта развития: заявка либо одобряется подходящим ответственным лицом, либо передаётся на согласование другому человеку, ответственому за заявки на более крупные суммы кредита.

На рисунке 2 представлена часть общего бизнес-процесса, которая связана с генерацией документов по шаблону.

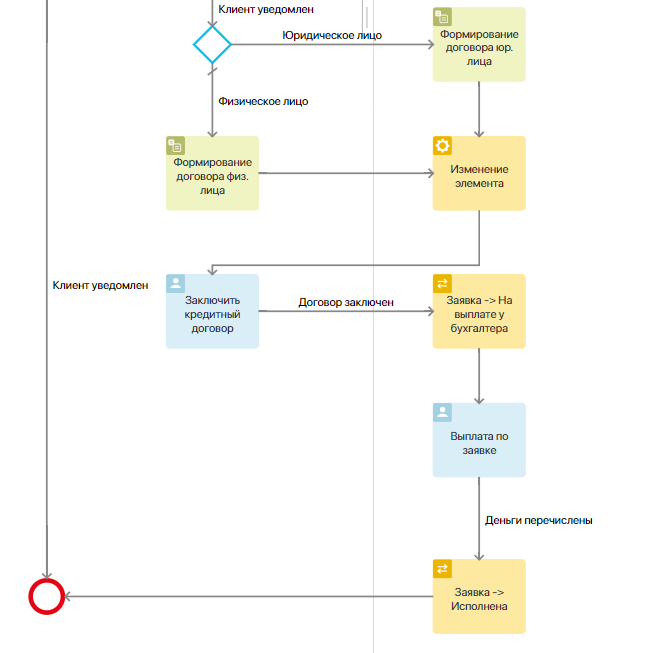


Рисунок 2 – Часть бизнес-процесса

Для того, чтобы при заключении договора с клиентами формировался договор, была реализована часть бизнес процесса, связаная с генерацией документа по шаблону.

Исключающий шлюз «или/или» добавлен для того, чтобы проверялось условие кто заключает договор на кредит (физическое лицо/юридическое).

На рисунке 3 показана настройка условий перехода к следующим блокам процесса.

В начале бизнес-процесса был использован блок с присваиванием контекстных переменных, в котором была занесена переменная «Юридическая форма заявителя», благодарая чему может производится проверка на тип формы заявителя и будет выбран правильный шаблон для генерации договора.

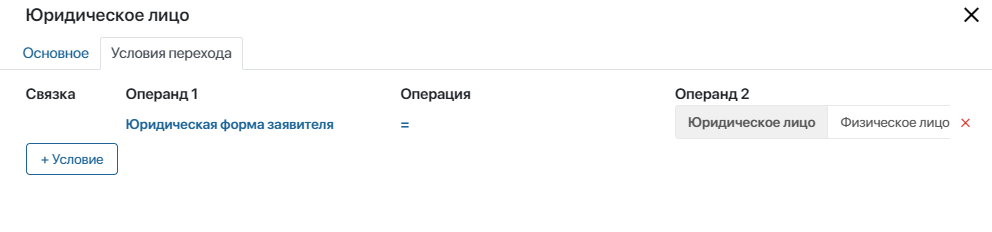


Рисунок 3 – Настройка условия перехода к формированию договора по кредитной заявке

Были добавлены блоки с генерацией договора по шаблону для обеих типов форм заявителя. Для их настройки был создан шаблон договора так, чтобы вся необходимая информация о клиенте подтягивалась из приложений. На рисунке 4 представлен шаблон формирования договора.

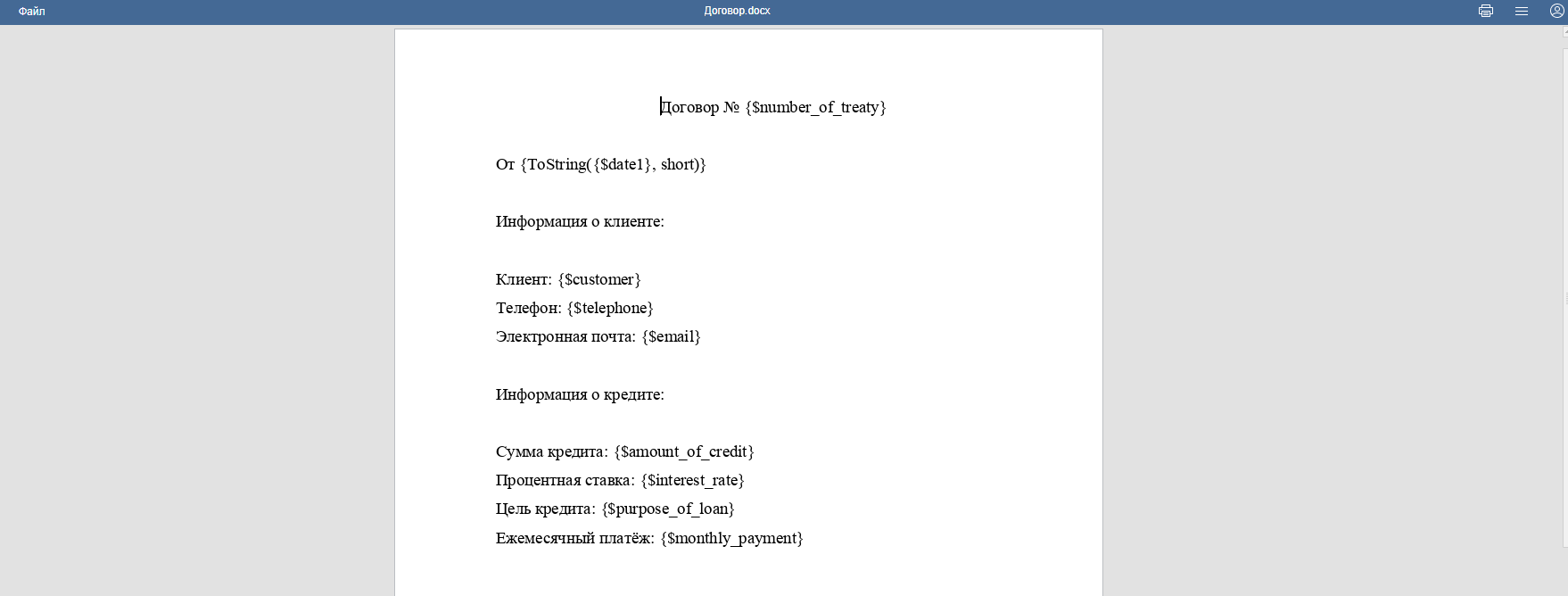


Рисунок 4 – Шаблон формирования договора

На рисунке 5 показана настройка значения полей, где показана связь переменной из шаблона с контекстной переменной в приложении.

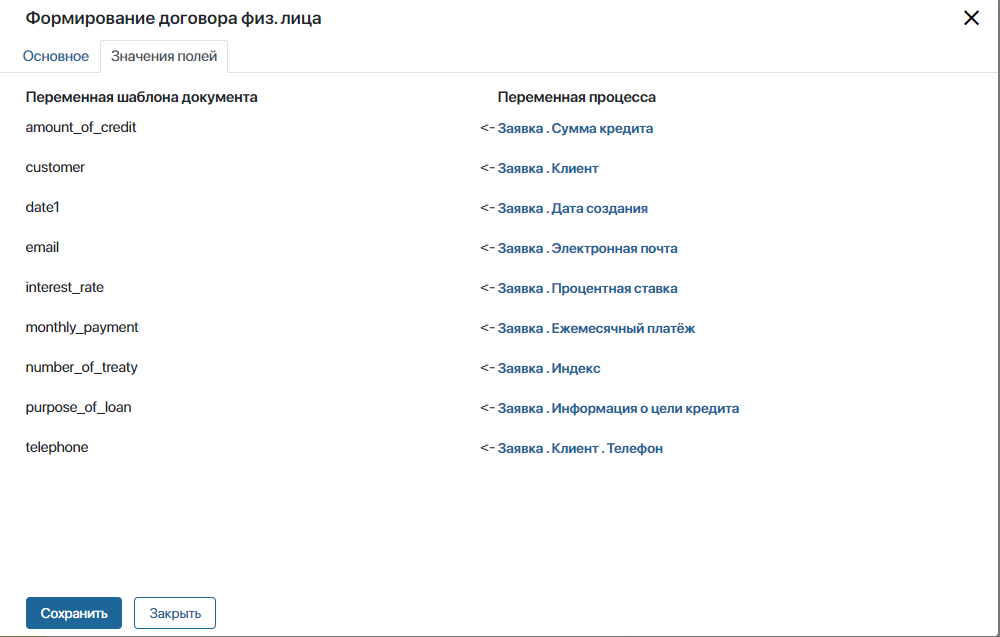


Рисунок 5 – Настройка значения полей

Аналогичным образом был настроен блок с формированием договора для юридического лица.

Далее был добавлен блок «Изменение элемента», он используется для редактирования полей в карточке элемента приложения в ходе выполнения бизнес-процесса.

На рисунке 6 показана настройка значения полей при изменении договора. Так как при создании договора могут измениться данные о клиенте. Договор будет хранится в карточке клиента и его можно посмотреть.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6 – Настройка значений полей для блока «Изменение элемента»

Были использованы блоки задач. Они нужны для постановки задачи определенной зоне ответственности.

На рисунке 7 представлена настройка блока задачи «Заключить кредитный договор». Если договор будет согласован, то будет поставлена новая задача и статус кредитной заявки также будет изменен.

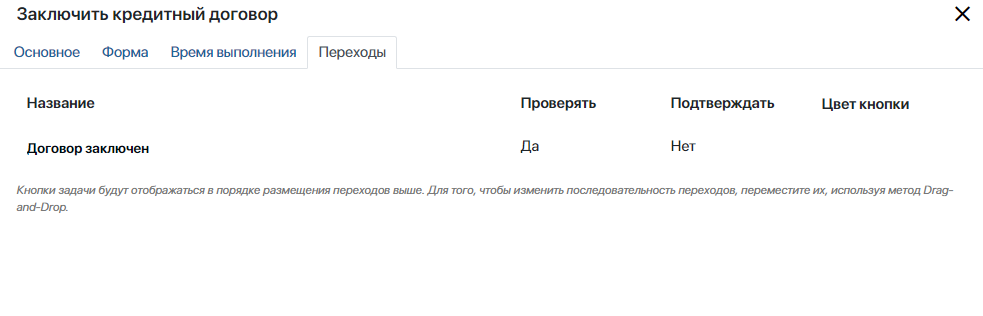


Рисунок 7 – Настройка блока задачи «Заключить кредитный договор»

Аналогичным образом был настроен еще один блок задачи «Выплата по заявке».

В бизнес-процессе были использованы блоки «Управление статусом» для того, чтобы по мере выполнения задач в разных зонах ответственности менялся статус кредитной заявки. На рисунке 8 показана настройка блока «Управление статусом». После того как сотрудник отдела продаж выполнит свою задачу и заключит кредитный договор, то статуст заявки будет «На выплате у бухгалтера».

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9 – Настройка блока «Управление статусом»

На рисунке 10 представлен бизнес-процесс «Анализ клиентов».

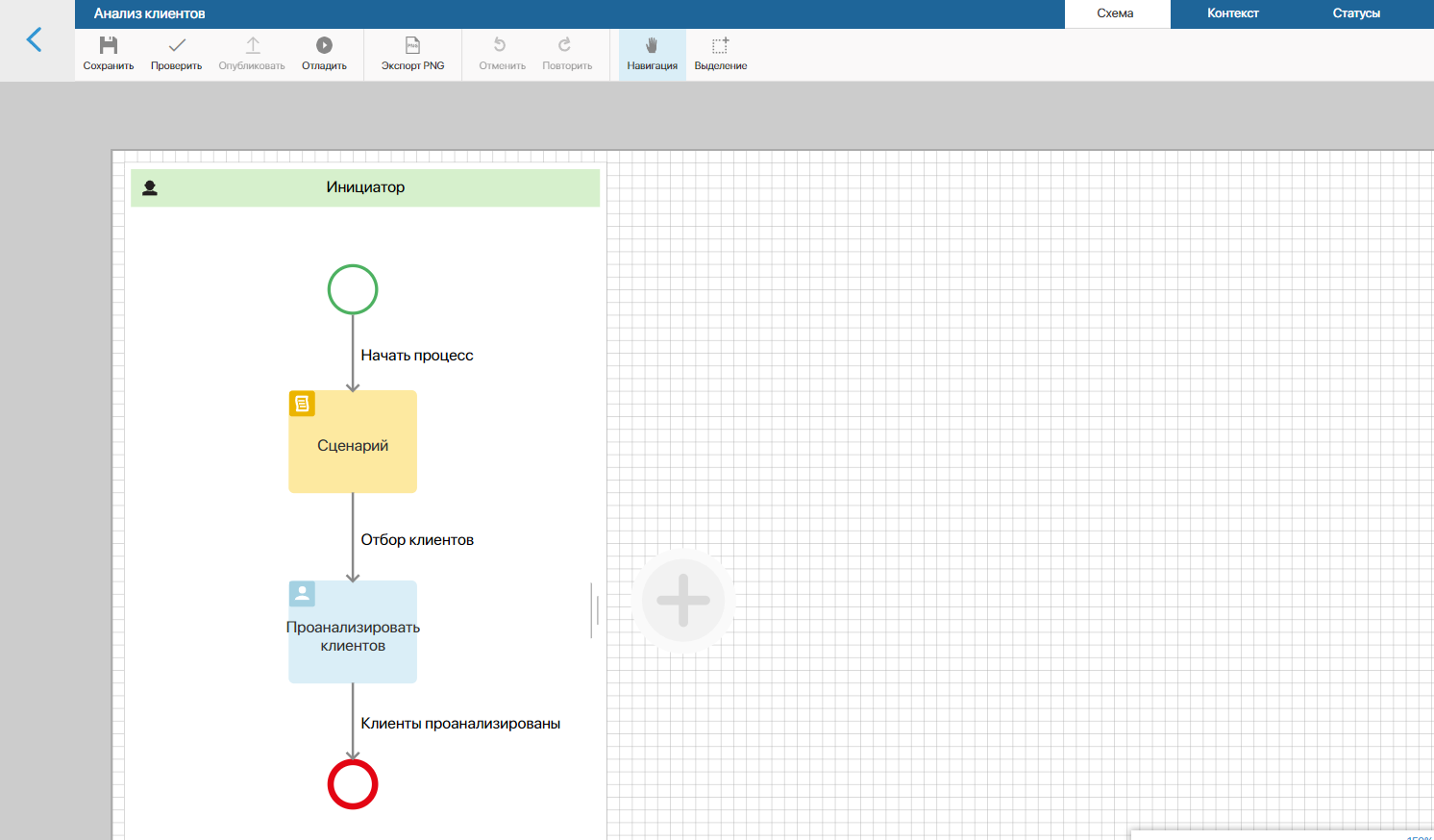


Рисунок 10 – Бизнес-процесс «Анализ клиентов»

Данный бизнес-процесс был реализован для проверки и анализа клиентов, которые имееют больше, чем одну кредитную историю. При запуске процесса, с помощью кнопки «Анализ клиента» в приложении «Клиенты» происходит проверка всех клиентов системы, и выводится на экранной форме ФИО тех, на чье имя было оформлено большего одного кредита.

На рисунке 11 представлено начало бизнес-процесса аналища клиентов.

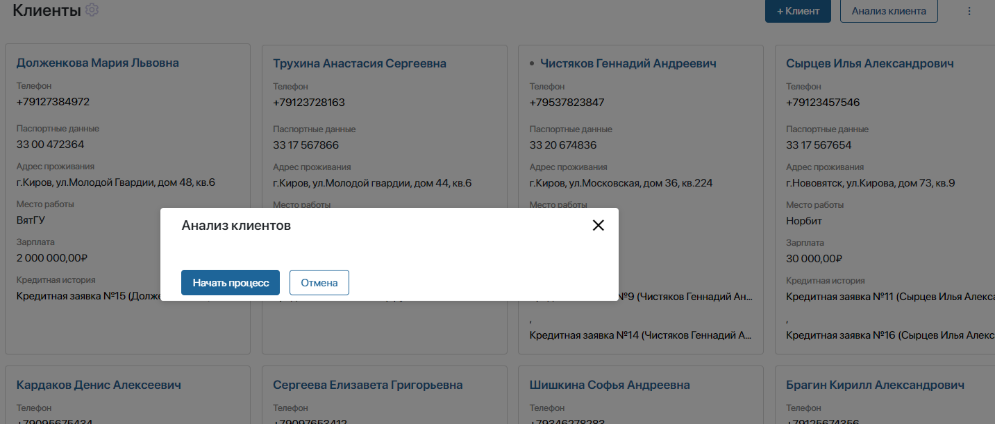


Рисунок 11 – Начало процесса анализа клиентов

На рисунке 12 прдеставлен результат анализа клиентов, выведены ФИО тех, кто оформил кредит больше одного раза.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 12 – Результат анализа клиентов

Нажав на любого клиента в списке, можно перейти на вкладку того самого клиента и посмотреть основную информацию о нем и кредитную историю.

На рисунке 13 показан код, как происходил анализ клиентов с кредитной историей > 1.

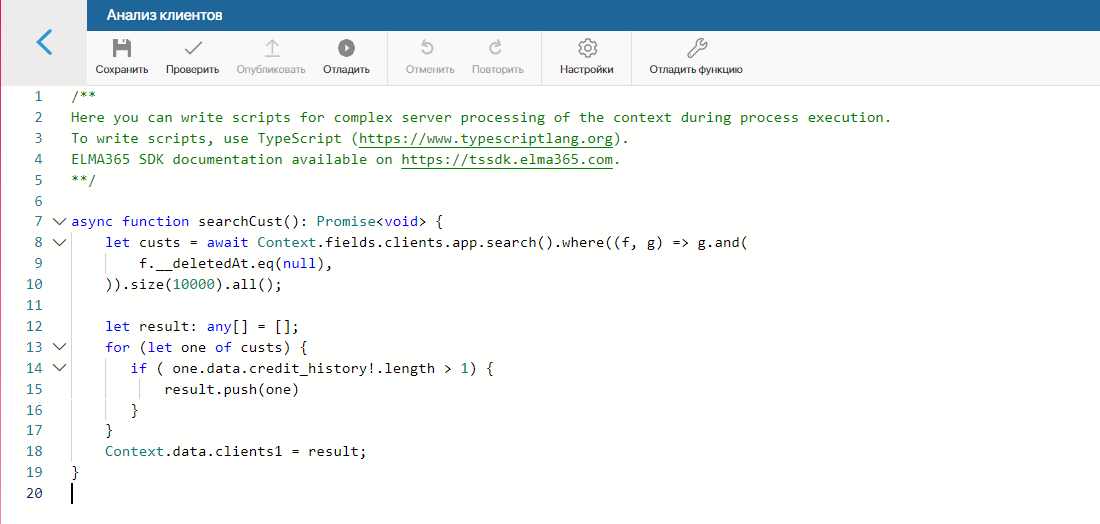


Рисунок 13 – Код анализа клиентов с кредитной историей > 1

В листинге программы представлены все функции, необходимые для работы системы. Функция ChangeonClient отвечает за отображение или скрытие полей «Телефон» и «Электронная почта» у физического лица в зависимости от введённых данных. Функция changeType отвечает за смену формы заявителя: юридическое или физическое лицо. Функция checkIncome отвечает за проверку того, что расходы юридического лица не превышают его доходы. Функция changeOnPred также отвечает за отбражение или скрытие полей «Телефон» и «Электронная почта» у представителя юридического лица.

ЛИСТИНГ ПРОГРАММЫ

**«Форма кредитные заявки»**

/\* Client scripts module \*/

async function onInit(): Promise<void> {

    ViewContext.data.hide = true

    ViewContext.data.hide\_tab = true

    ViewContext.data.cancel\_window = true

}

async function changeonClient(): Promise<void> {

    let client = await Context.data.customer!.fetch()

    Context.data.telephone = client.data.telephone

    Context.data.email = client.data.email

    ViewContext.data.hide = false

}

async function changeType(): Promise<void> {

    if (Context.data.legal\_form\_of\_the\_applicant == true) { // юр.лицо

        ViewContext.data.hide\_tab = false

        ViewContext.data.variable = undefined

    }

    else {

        ViewContext.data.hide\_tab = true

        ViewContext.data.variable = "текст"

    }

}

async function checkIncome(): Promise<void> {

    let monthly\_income = Context.data.monthly\_income

    let monthly\_expenses = Context.data.monthly\_expenses

    let income = monthly\_income!.asFloat()

    let expenses = monthly\_expenses!.asFloat()

    if (income < expenses) {

        ViewContext.data.cancel\_window = false

        ViewContext.data.variable = undefined

    }

    else {

        ViewContext.data.cancel\_window = true

        ViewContext.data.variable = "текст"

    }

}

async function changeOnPred(): Promise<void> {

     let pred = await Context.data.representative\_of\_the\_legal\_entit!.fetch()

    Context.data.telephone = pred.data.telephone

    Context.data.email = pred.data.email

    ViewContext.data.hide = false

}