**Я как клиент банка хочу создать заявку на открытие счета**

**История версий**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Задача** | **Тип изменения** | **Цвет изменения** | **Дата** | **Автор** |
| Ссылка на задачу | Создано |  | 23.05.2024 | Тинькова Е |

**Бизнес- анализ**

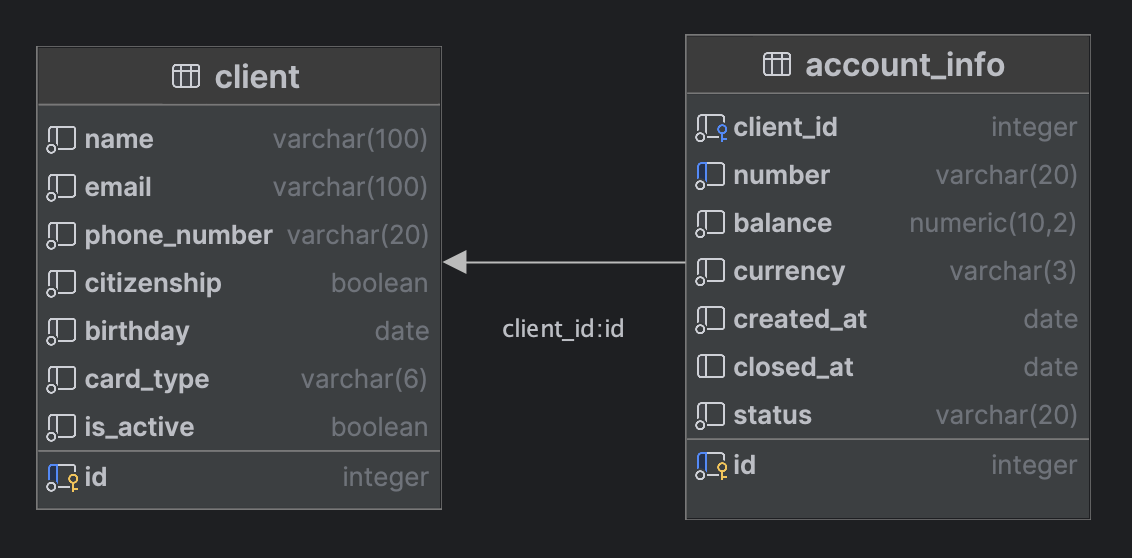
- ссылка

**Назначение и цель**

Создание заявки на открытие счета после заполнения клиентом анкеты на сайте/в мобильном приложении

**Предварительные действия**

- Реализована модель данных в СУБД:



- Пользователь перешел на страницу <https://our-bank.ru/cards> или открыл в мобильном приложении экран с оформлением нового счета

- Пользователь выбрал тип счета (кредитный/дебетовый) и корректно ввел информацию о себе в обязательные поля анкеты

- Произведена верификация номера или пользователь прошел аутентификацию и авторизацию в приложении, нажал кнопку «Оформить»

**Ограничения по ролям**

Доступ к интерфейсу возможен для пользователя с ролью CLIENT ALL

**Макеты**

- ссылка на Figma/Pixco/draw.io итд

**Основной сценарий (успешный сценарий)**

1. Front/MobileApp отправляет запрос на создание заявки по открытию нового счета

на Back (в систему обработки анкет) POST /filledForms (ссылка на API)

1. Система обработки анкет получает данные, создает заявку на открытие счета и возвращает успешный ответ Front.
   1. Если система обработки анкет вернула на Front/MobileApp ошибки 4хх, 5хх см. Альтернативный сценарий 1
2. Front/MobileApp отображает пользователю информацию о дальнейших действиях и сроках ожидания карты, контактах поддержки и адресах отделений банка с п.Макеты

**Альтернативный сценарий**

1. Front/MobileApp отображает ошибку от Back с п.Макеты

**Требования к интерфейсу**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Название атрибута** | **Тип** | **Длина** | **Редактируемость** | **Маска ввода /отображения** | **Обязательность** | **Сортировка** | **Значение по умолчанию** | **Источник данных** | **Примечания** |
| ФИО пользователя | Label (поле) | 100 | + | - | + | - | - | Пользователь | Пример: Иванов Иван Иванович |
| Email | Label (поле) | 100 | + | example@domain.ru | + | - | - | Пользователь | Пример: example@domain.ru |
| Номер телефона | Label (поле) | 20 | + | +7 (xxx) xxx-xx-xx | + | - | +7 | Пользователь | Пример: +7 (901) 123-45-67 |
| Гражданство РФ | Button group | - | - | - | + | - | Да | Пользователь | Возможные варианты: Да/Нет |
| Дата рождения | Label (поле) | - | + | DD.MM.YYYY | + | - | - | Пользователь | Пользователь должен быть старше 14 лет |
| Тип счета | Button group | - | - | - | + | - | DEBET | Пользователь | Возможные варианты: DEBET/CREDIT |

**Результат**- Система обработки анкет сформировала заявку на открытие счета для клиента и передала её в систему обработки заявок (см. задачу «Создание счета», ссылка)

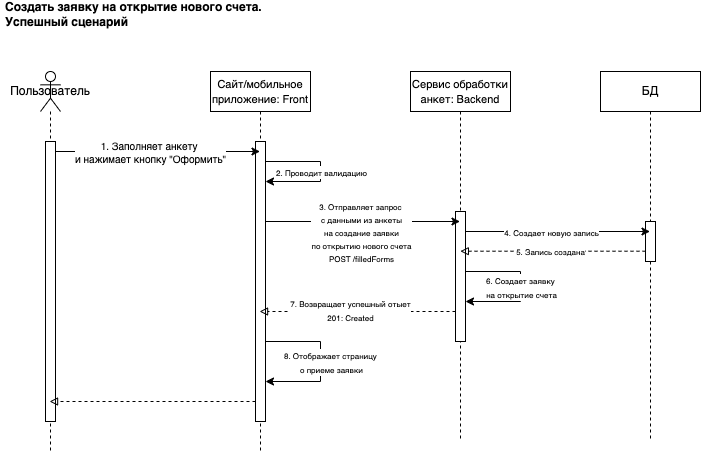
**POST /filledForms Создать новую заявку на открытие счета**

**История версий**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Задача** | **Тип изменения** | **Цвет изменения** | **Дата** | **Автор** |
| Ссылка на задачу | Создано |  | 23.05.2024 | Тинькова Е |

**Сервис**Микросервис application-request (Система обработки анкет)

**Алгоритм вызова**

****

**Свойства вызова**

|  |  |
| --- | --- |
| **Свойство** | **Описание** |
| Тип взаимодействия | **REST**/Kafka/Артемис MQ/gRPC/GraphQL |
| Формат запроса/ответа | **JSON**/XML |
| Тип вызова | **Синхронный**/Асинхронный |
| Протокол взаимодействия | **HTTPS**/MTLS |
| Способ аутентификации | Bearer Token/ API key/ **JWT** |

**Формат запроса**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Параметр** | **Тип** | **Длина** | **Расположение** | **Обяз** | **Описание** |
| - | - | - | - | - | - |

**Пример**POST/filledForms

**Тело запроса**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Параметр** | **Тип** | **Длина** | **Расположение** | **Обяз** | **Описание** | **Место для сохранения** |
| name | string | 100 | body | + | ФИО клиента | client.name |
| email | string | 100 | body | + | Эл. почта клиента | client.mail |
| phoneNumber | string | 20 | body | + | Номер телефона клиента | client.phone\_number |
| citizenship | boolean | - | body | + | Гражданство РФ (true/false) | client.citizenship |
| birthday | date | - | body | + | Дата рождения | client.birthday |
| cardType | string | 6 | body | + | Тип счета: CREDIT/DEBET | client.card\_type |
| isActive | boolean | - | body | + | Статус клиента: true – активен, false – не активен | client.is\_active |

**Пример**

{

"name": "Иванов Иван Иванович",

"email": "example@domain.ru",

"phoneNumber": "+7 (901) 123-45-67",

"citizenship": true,

"birthday": "01.01.1976",

"cardType": "DEBET",

"isActive": true

}

**Формат ответа**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Параметр** | **Тип** | **Длина** | **Обязательность** | **Описание** | **Источник данных** | **Отображение на фронте** |
| id | integer | - | + | Идентификатор созданной заявки | client.id | - |

**Пример**{

"id": "256432"

}

**Коды ответов**

|  |  |  |  |
| --- | --- | --- | --- |
| **Код** | **Название** | **Описание** | **Поведение системы** |
| 201 | OK | Заявка создана | Продолжить процесс |
| 400 | Bad Request | Некорректный запрос | Отобразить ошибку |
| 429 | Too many requests | Слишком много запросов | Отобразить ошибку |
| 500 | Internal Server Error | Внутренняя ошибка сервера | Отобразить ошибку |

**Алгоритм обработки запроса**

1. Так как валидация данных уже произведена на Front, Back при получении запроса обращается в таблицу client и создает запись о новой заявке
2. Пример SQL-запроса:

INSERT INTO client (name, email, phone\_number, citizenship, birthday, is\_active)

VALUES (

json\_data->>'name',

json\_data->>'email',

json\_data->>'phoneNumber',

(json\_data->>'citizenship')::boolean,

TO\_TIMESTAMP(json\_data->>'birthday', 'DD.MM.YYYY'),

(json\_data->>'isActive')::boolean

);

1. После успешной вставки в таблицу Back возвращает успешный код ответа на Front
2. Информация о заявке передается в систему создания счетов

**Альтернативные процессы**

Ошибки 4хх, 5хх

1. Back вернуть ошибку с текстом на Front