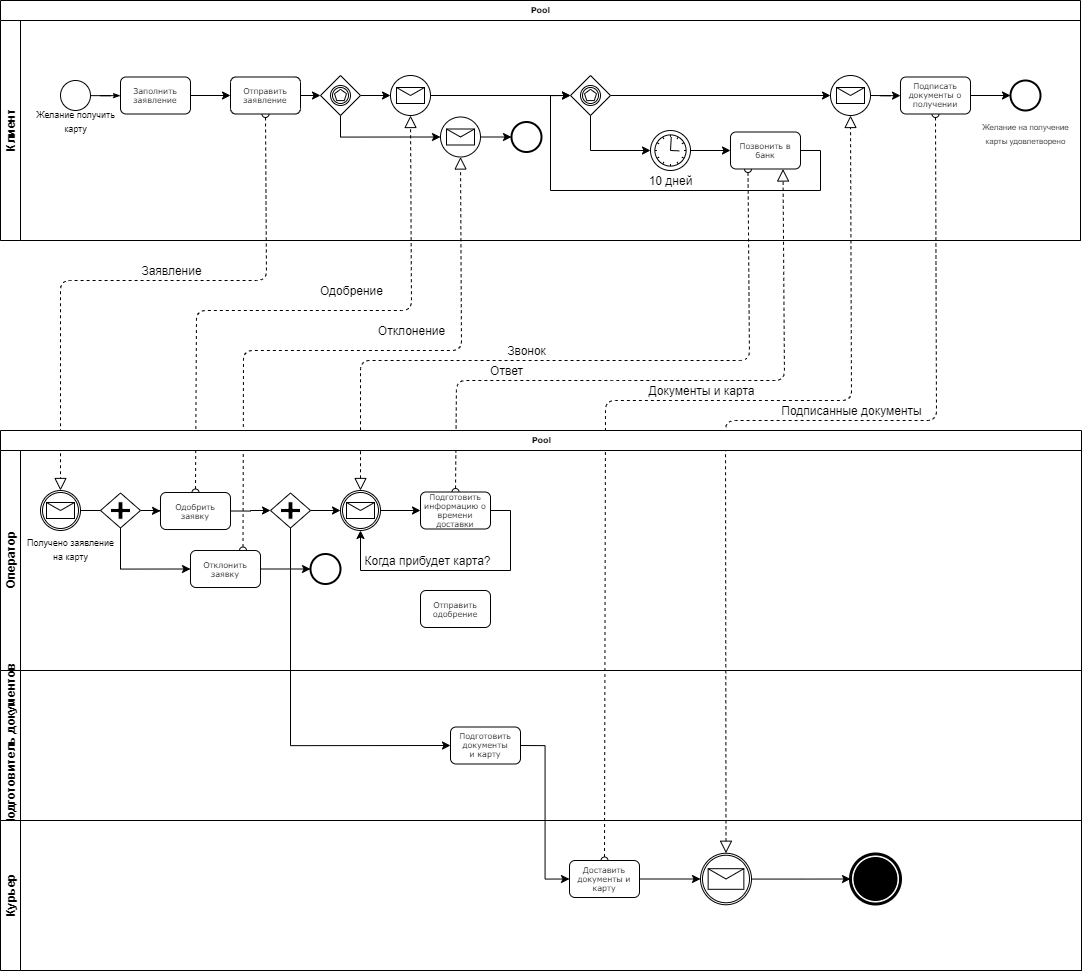
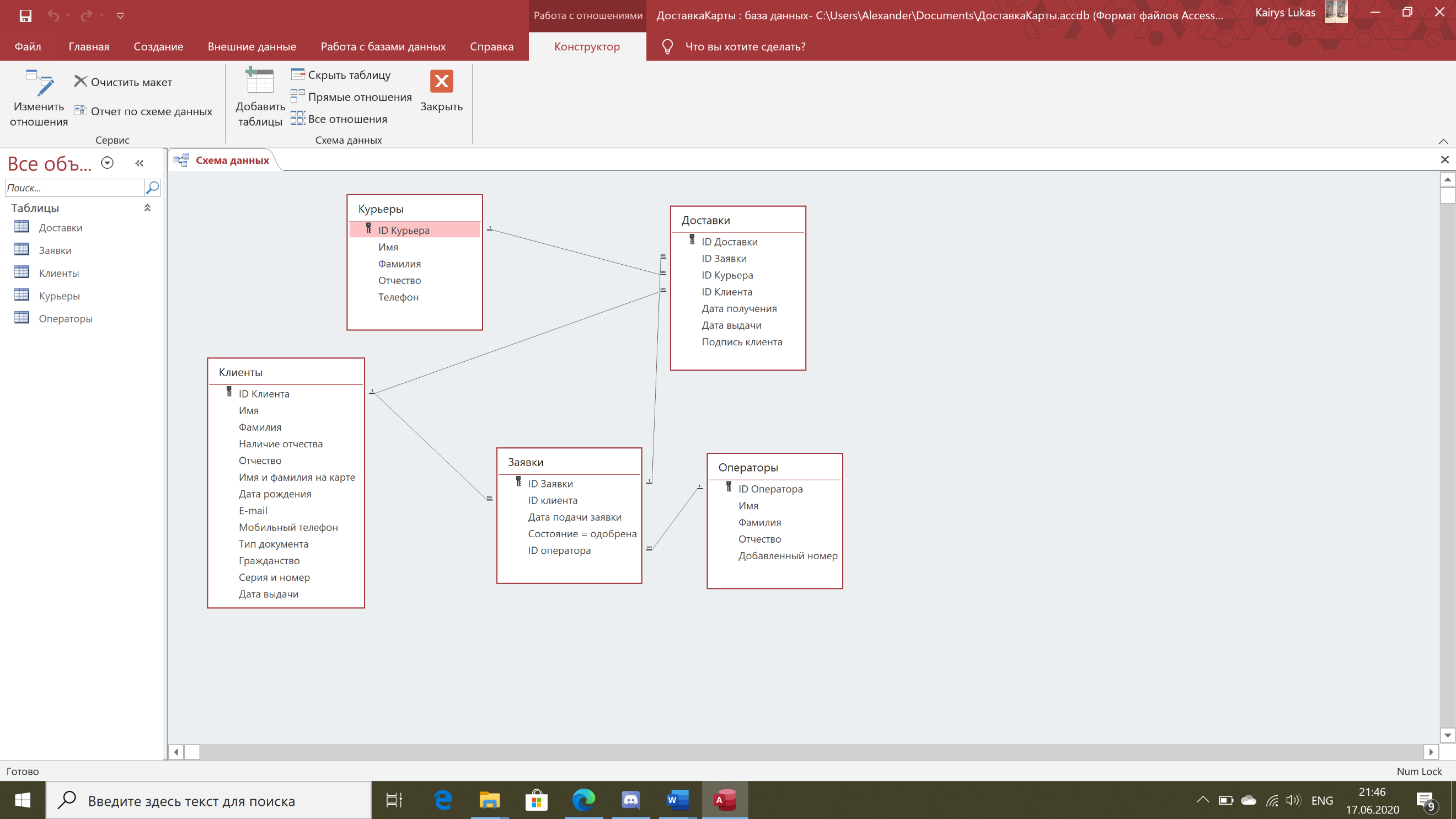
|  |  |
| --- | --- |
| **Задание** | Создать мобильное приложение (iOS/Android) для доставки банковских карт клиенту. Приложение должно автоматизировать все этапы доставки от приема заявки до вручения и получения обратной связи. Предоставить клиенту прозрачную картину прохождения каждого этапа. |
| **Автор решения** | Григорьев Александр Дмитриевич |

Описание процесса получения банковской карты клиентом приведено в виде диаграммы в нотации BPMN 2.0.



На основании анализа бизнес-процессов построим информационную модель системы в виде схемы данных.



Проектирование клиентского приложения

Целевая функция: автоматизировать процесс подачи заявки и обратной связи.

Требования к функциональным характеристикам

1) Входные и выходные данные

А) Входная информация: информация о клиенте

Б) Выходная информация: статусы работы

2) Основные функции приложения:

* Заполнение и отправка заявки
* Просмотр текущего статуса заявки

Архитектура системы

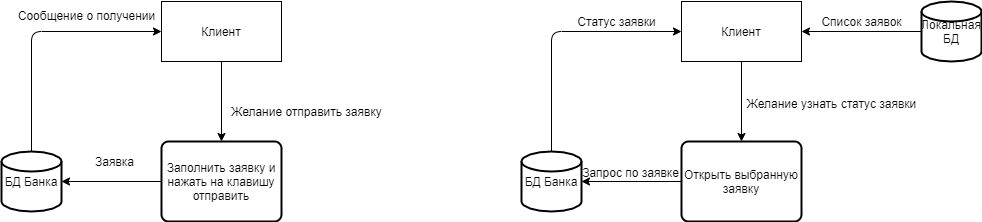
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Клиент | | Облачная БД | Банк (Сервер) | |
|  |  |  |  |  |
| Локальная БД | **Интерфейс клиента** | **FireBase RealTime DataBase** | **Интерфейс оператора** | **БД банка** |

Так как главным требованием является создание «клиентоориентированного» приложения, то есть клиентского, следовательно, для простоты реализации БД банка будет храниться на устройстве оператора.

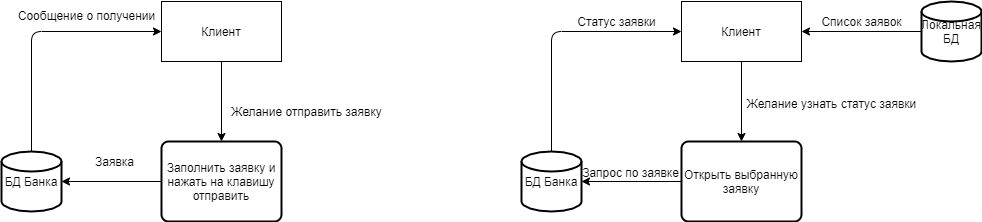
Проектирование функционала

Методы реализации функций отображены с помощью DFD-диаграмм

1. Заполнение и отправка заявки



1. Просмотр текущего статуса заявки



Разработка и реализация интерфейсов

1. Клиентское приложение CardDelivery

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Главное меню | **Оформление заявки** | **Список заявок** | **Статус заявки** |

1. Серверный GUI CardDeliveryOperator

|  |  |
| --- | --- |
|  |  |
| Входящие заявки | **Просмотр и изменения статуса заявки** |

Дополнительно

Приложение также настроено на прием уведомлений в закрытом состоянии с помощью FirebaseMessagingService. С помощью специального интерфейса Cloud Messaging от Firebase.

