

Министерство образования Республики Беларусь

Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И  
РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления  
Кафедра интеллектуальных информационных технологий

ОТЧЁТ  
по лабораторной работе №2  
по дисциплине

ПРОЕКТИРОВАНИЕ БАЗ ЗНАНИЙ

Вариант №4

Студент гр. 221703

Д.Н. Вечорко

Руководитель

И.А. Ерофеев

Минск 2024

# 1 ПОСТАНОВКА ЗАДАЧИ

## 1.1 Словесное описание предметной области

Технолог междугородней телефонной станции (МТС) вносит в БД информацию о новых клиентах, пользующихся услугами, это: номер телефона клиента, ФИО, адрес, дата регистрации. Также технолог изменяет цены за пользование услугами, эти данные хранятся в следующем виде: дата, название населенного пункта, с которым осуществляется связь, стоимость одной минуты разговора, льготная стоимость (с 20.00 до 6.00). Оператор МТС фиксирует дату разговора, город, с которым произошел разговор, номер абонента, длительность разговора. После этого оператор отправляет абоненту квитанцию об оплате. Он также принимает извещение об оплате и фиксирует, что те или иные разговоры оплачены.

## 1.2 Требования к приложению

Необходимо реализовать выполнения следующих функций:

- Добавление/редактирование/удаление информации о клиентах телефонной станции.
- Добавление/редактирование/удаление информации о стоимости услуг связи.
- Добавление/редактирование/удаление информации о телефонном звонке (функция оператора МТС).
- Расчет количества абонентов, соединившихся с выбранным городом за выбранный месяц с разбивкой по дням.
- Просмотр списка абонентов, имеющих задолженности по оплате (срок оплаты – 20 дней) – дата, название организации, предоставляющей услуги связи, телефон, ФИО, адрес абонента.
- Вывести для просмотра стоимость оплаты одной минуты разговора для разных населенных пунктов на заданную дату: название организации, предоставляющей услуги связи - дата, название населенного пункта, стоимость одной минуты, льготная стоимость одной минуты.

## 2 КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ

### 2.1 Варианты использования

Система имеет два основных пользователя – технолог и оператор МТС.

Технолог вносит в базу данных информацию о новом клиенте (номер телефона, ФИО, адрес, дата регистрации). Также технолог обновляет данные о тарифах, включая дату, населённый пункт, стоимость одной минуты разговора, и льготную стоимость.

Оператор МТС фиксирует информацию о разговоре записывая дату, город, номер абонента и длительность разговора. Кроме того оператор может отсылать квитанцию об оплате, а также получать подтверждение оплаты от абонента и фиксировать, что разговор оплачен. (рис. 2.1).



Рисунок 2.1 – Диаграмма вариантов использования

### 2.2 Отношения сущностей

В базе данных определено четыре таблицы: стоимость, клиент, звонок и квитанция.

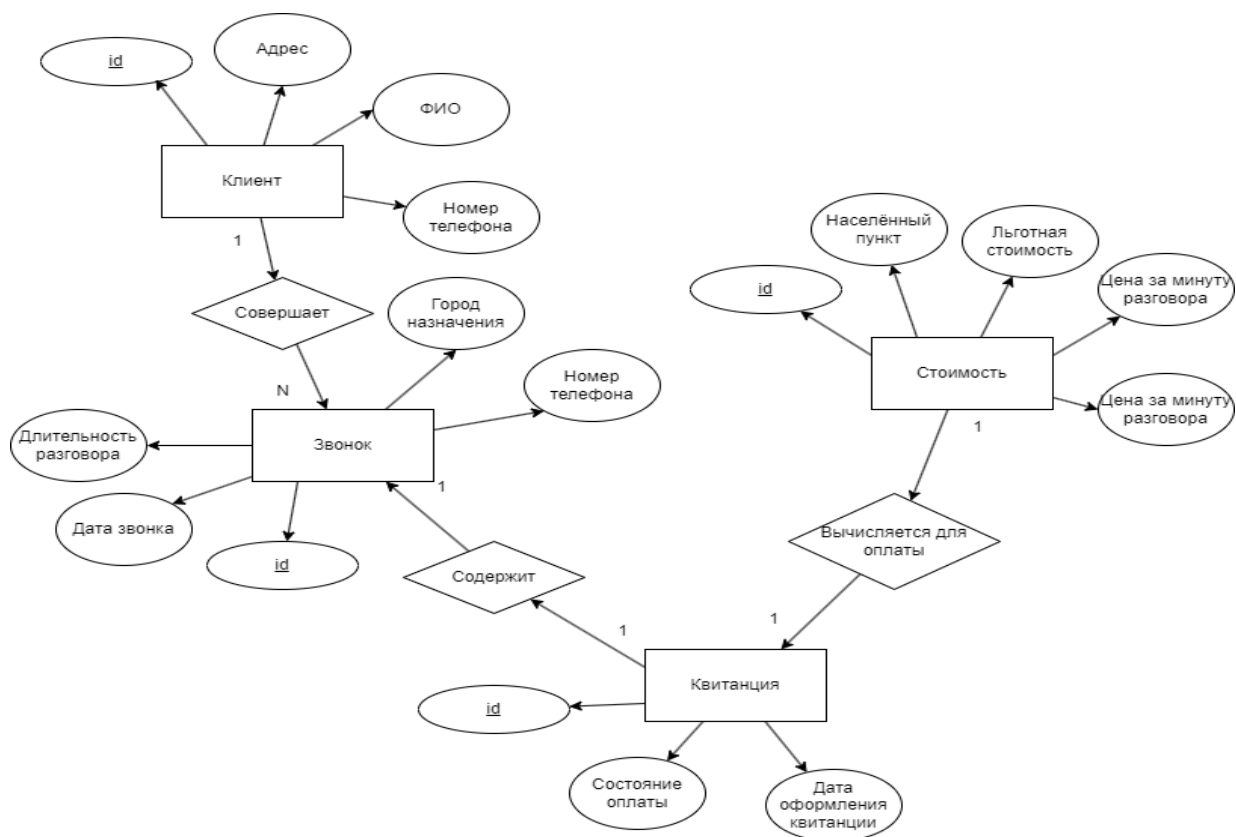


Рисунок 2.2 – ER(Л)-диаграмма

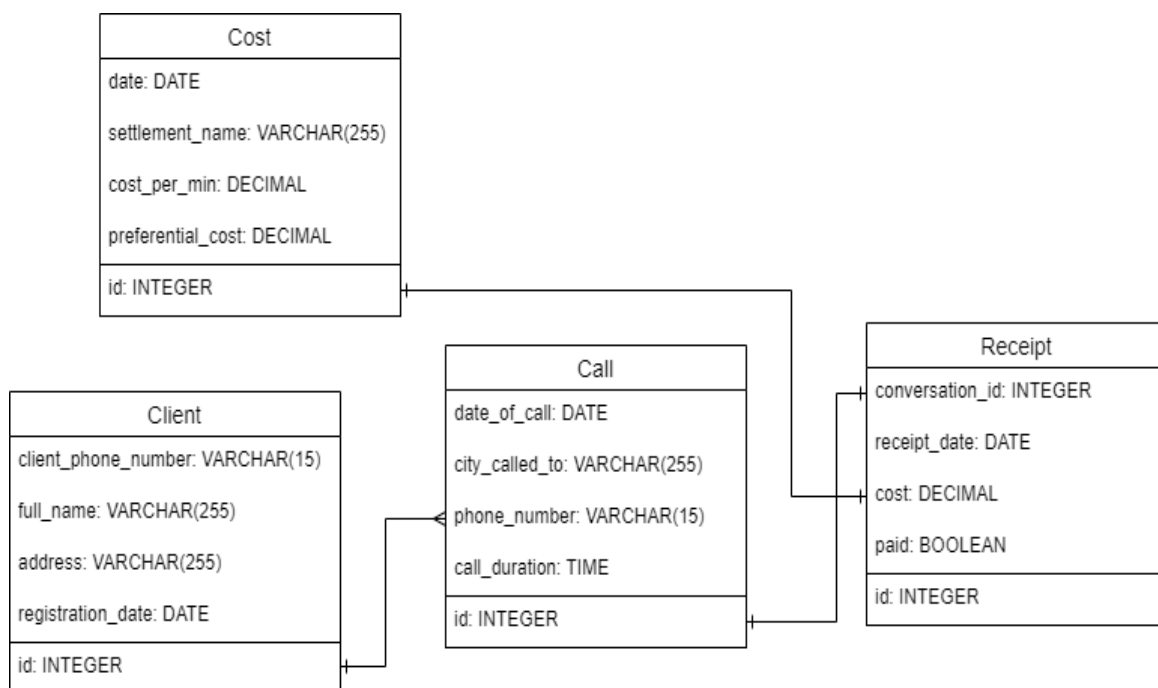


Рисунок 2.3– ER(Ф)-диаграмма

### 3 ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

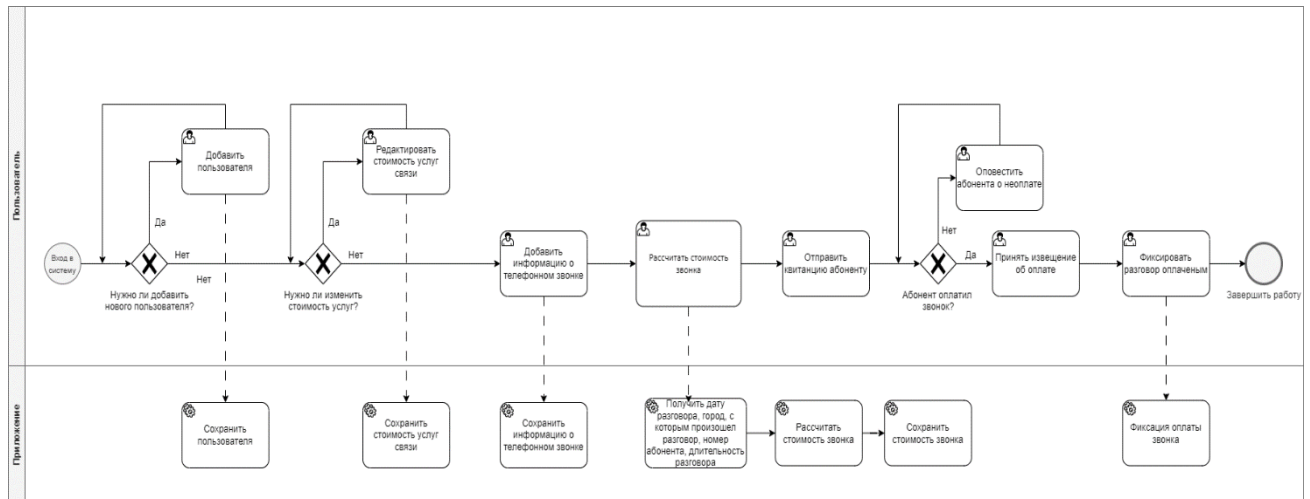


Рисунок 3.1 – BPMN-диаграмма

### 4 ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

#### 4.1 Схема базы данных

##### 4.1.1 Таблица стоимости

```
CREATE TABLE Cost( id INT PRIMARY KEY GENERATED ALWAYS
AS IDENTITY,
date: DATE,
settlement_name: VARCHAR(255) NOT NULL,
cost_per_min: DECIMAL NOT NULL,
preferential_cost: DECIMAL
);
```

##### 4.1.2 Таблица клиентов

```
CREATE TABLE Client( id INT PRIMARY KEY GENERATED ALWAYS
AS IDENTITY,
date: DATE,
full_name: VARCHAR(255) NOT NULL,
address: VARCHAR(255),
registration_date: DATE NOT NULL
);
```

##### 4.1.3 Таблица звонков

```
CREATE TABLE Call( id INT PRIMARY KEY GENERATED ALWAYS
AS IDENTITY,
date_of_call: TIMESTAMP
city_called_to: VARCHAR(255) NOT NULL,
phone_number: VARCHAR(15) NOT NULL,
call_duration: TIME NOT NULL
UNIQUE (client_id, date_of_call), FOREIGN KEY
(client_id) REFERENCES Client(id)
);
```

#### 4.1.4 Таблица квитанций

```
CREATE TABLE Receipt( id INT PRIMARY KEY GENERATED
ALWAYS AS IDENTITY,
conversation_id: INTEGER,
receipt_date: DATE,
cost: DECIMAL NOT NULL,
paid: BOOLEAN NOT NULL
);
```

5

6

## 7 ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

### 7.1 PostgreSQL

В качестве базы данных для реализации приложения была выбрана PostgreSQL - одна из наиболее популярных реляционных баз данных на данный момент. Её преимуществами являются открытый исходный код, большое сообщество, возможность интеграции с большинством современных языков программирования, а также наличие качественной документации.

### 7.2 Docker, Docker Compose

Для удобства запуска и конфигурации PostgreSQL использовалась технология контейнеризации Docker, которая позволяет быстро и надежно запускать любые программные системы в любых условиях. Преимуществами Docker является кросс-платформенность и большое количество обучающих материалов. При помощи Docker возможно быстро установить и настроить PostgreSQL (или любое иное программное обеспечение).

### 7.3 Java

Для разработки интерфейса приложения использовался язык Java. Выбор данного языка программирования обусловлен его высокоуровневостью, удобством в разработке и поддержке, наличии большого количества обучающих материалов по нему в сети Интернет.