Содержание:

1. Постановка задачи: словесное описание предметной области. (стр 3)
2. Концептуальное проектирование: Диаграммы вариантов использования. (стр 4)
3. Логическое проектирование: Диаграмма сущность-связь. (стр 5)
4. Физическое проектирование: Схема БД. (стр 6)
5. Тексты основных запросов, функций, процедур и триггеров. (стр 7-8)
6. Интерфейс приложения: Основные окна с описанием. (стр 9-11)
7. Инструментальные средства, которые использованы при разработке. (стр 12)

**Постановка задачи: Словесное описание предметной области.**

**Вариант 4.** «Оплата за междугородние разговоры»

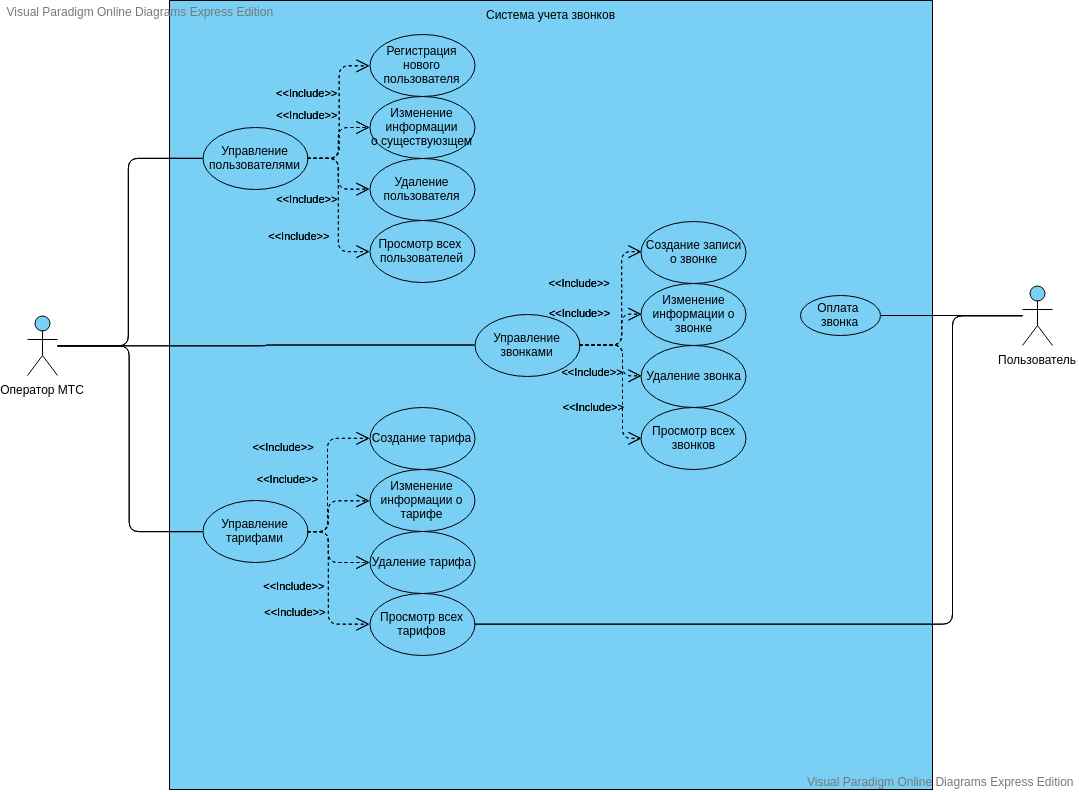
Предприятие связи, предоставляющие услуги междугородней связи.

**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 отсылает абоненту квитанцию об оплате. Он также принимает извещение об оплате и фиксирует, что те или иные разговоры оплачены.

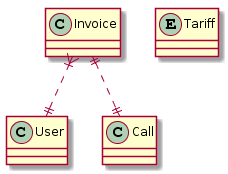
**Необходимо реализовать выполнения следующих функций**:

* 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.

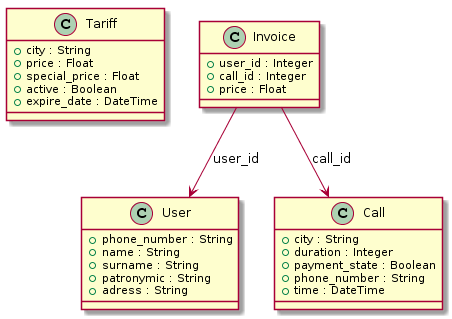
**Диаграммы вариантов использования (ВИ, Use Case)**



**Логическое проектирование: Диаграмма сущность-связь (ER-диаграмма)**

****

**Физическое проектирование: Схема БД.**

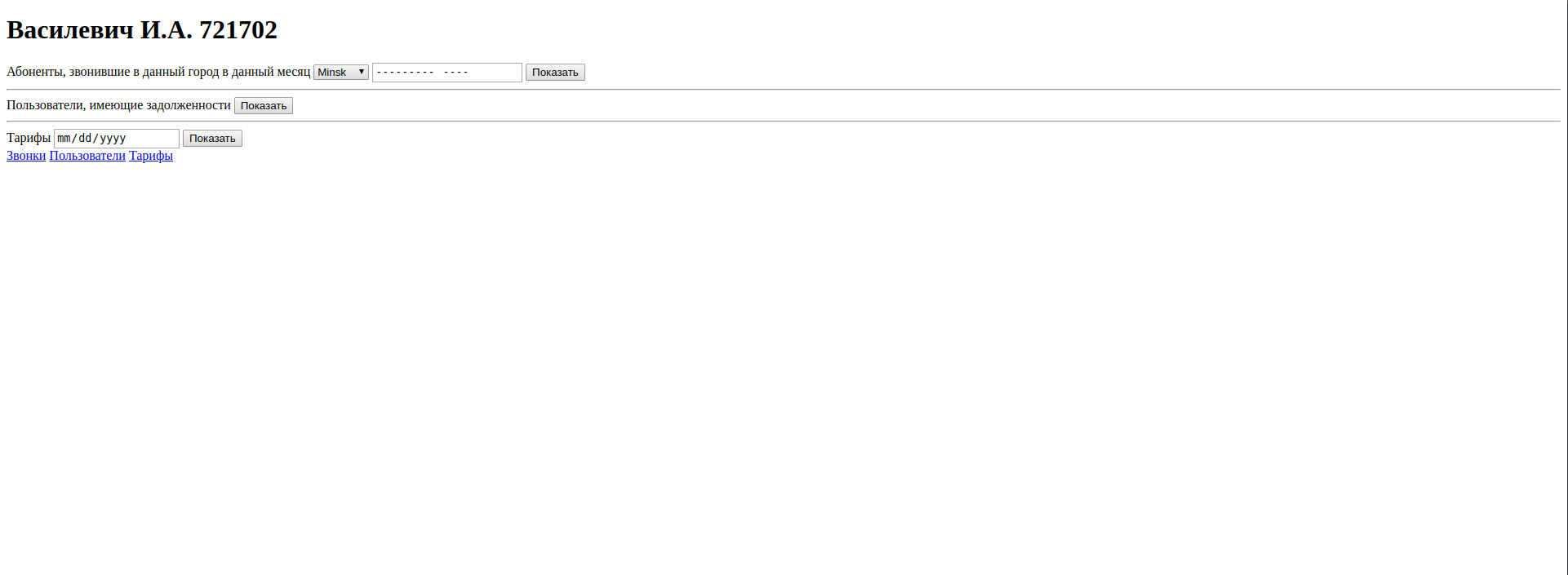


**Тексты основных запросов, функций, процедур и триггеров.**

Все запросы в базу данных выолнялись с помощью Все запросы в базу данных выполнялись с помощью механизма объектно-реляционного отображения. Поэтому для составления запросов использовался язык JPQL.

|  |  |  |
| --- | --- | --- |
| Текст запроса | Передаваемые аргументы | Результат запроса |
| **SELECT calls.\* FROM calls WHERE calls."city" = :city** | :city – название города | Возвращает объект, который соответствует звонку, совершенному в данный город |
| **SELECT \* FROM calls WHERE time <= :late\_time AND payment\_state = false** | :late\_time — время, при котором истекает срок оплаты звонка | Возвращает все звонки, которые не были оплачены в срок |
| **SELECT tariffs.\* FROM tariffs WHERE (expire\_date >= :today AND created\_at <= :today)** | :today – дата, в которую выполняется запрос | Возвращает тарифы, срок действия которых не истек |
| **SELECT \* FROM users WHERE users.id = :id** | :id – уникальный идентификатор пользователя | Возвращает объект, который соответствует представлению «пользователь» в БД |
| **SELECT \* FROM users** | - | Возвращает все объекты, соответствующие представлению «пользователь» в БД |
| **SELECT \* FROM tariffs WHERE tariffs.id = :id** | :id – уникальный идентификатор оборудования | Возвращает объект, который соответствует представлению «тариф» в БД |
| **SELECT \* FROM tariffs** | - | Возвращает все объекты, соответствующие представлению «тариф» в БД |
| **SELECT \* FROM calls** | - | Возвращает все объекты, соответствующие представлению «звонок» в БД |
| **SELECT \* FROM calls WHERE calls.id = :id** | :id – уникальный идентификатор оборудования | Возвращает объект, который соответствует представлению «звонок» в БД |

**Интерфейс приложения**



Страница 1. Список специальных операций оператора



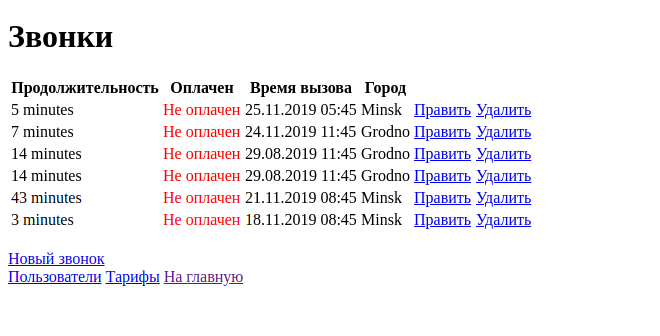
Страница 2. Разбивка звонков за месяц по дням



Страница 3. Страница с пользователями, имеющими задолженности



Страница 4. Тарифы на выбранную дату



Страница 5. Список всех звонков



Страница 6. Страница отдельного пользователя

**Инструментальные средства, которые использованы при разработке.**

Используемый язык программирования: Ruby 2.6.4

Используемые фреймворки: Ruby on Rails

Используемая СУБД: PostgreSQL

Используемые текстовые редакторы: VSCode, Vim

Используемая система контроля версий: git

Технологии создания веб-страниц: HTML, ERB