

ПО «ConnectionBase» ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Введение

При построении сетей предприятия при достаточной сложности топологии сети возникает необходимость в ведении учета, контроля и управления соединениями сети. Для автоматизации этих задач можно применить программное обеспечение.

Назначение

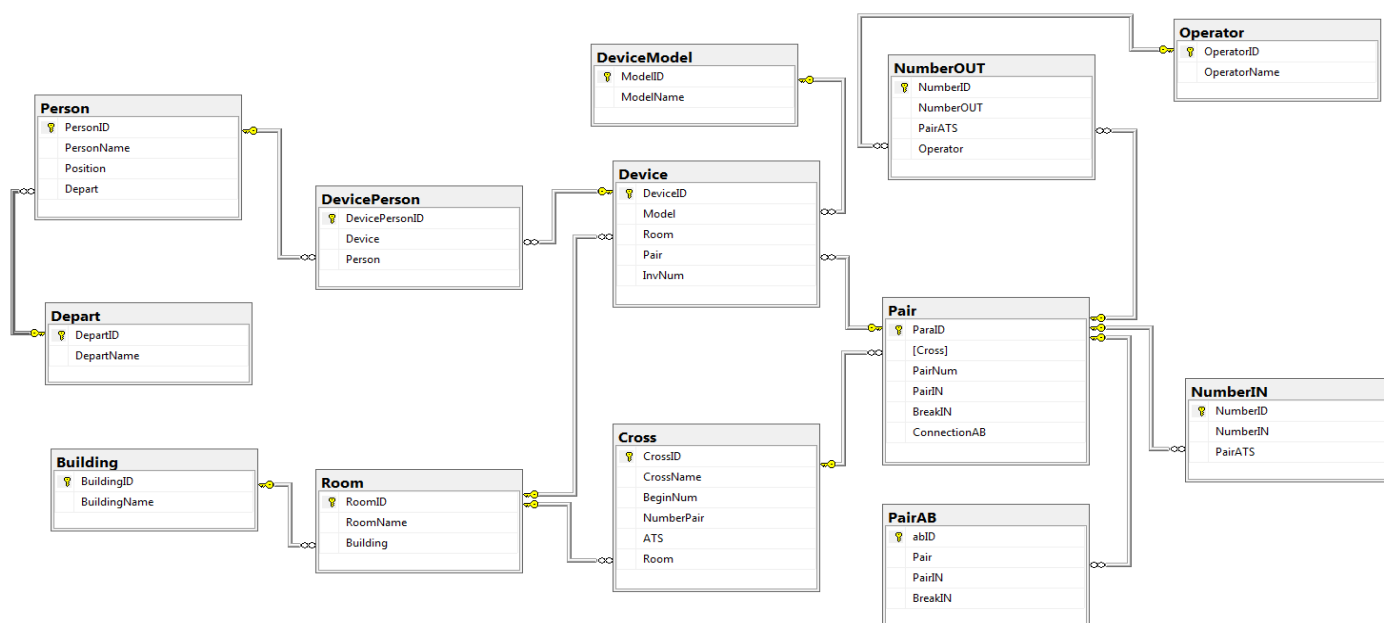
Данное ПО решает задачи учета на примере телефонной базы условного предприятия. Смысл учета заключается в отслеживании по соединениям в базе данных связи между активным оборудованием сети и абонентскими устройствами и построения последовательности соединений подключения, в том числе и при изменении соединений (частный случай для примера базы)

Характеристика объекта

Объект представляет собой комплекс зданий из трех корпусов, соединенных единой телефонной сетью. Телефонная сеть основана на базовом сервере, расположенном в Корпусе 1, и выносе, расположенном в Корпусе 2 (аналог «Alcatel-Lucent»). Схема сети расположена в файле «Схема сети.pdf», копия базы в файле

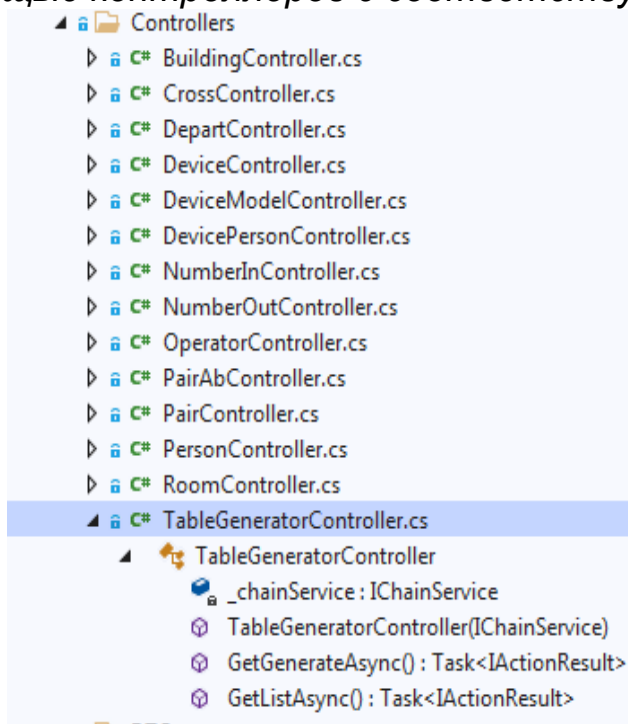
Основные решения

Структура базы:



ConnectionBase.API

Проверка входных данных осуществляется с помощью *Fluent Validation*. Обработка данных по таблицам осуществляется с помощью контроллеров с соответствующими названиями:



Получение результирующих таблиц для визуализации осуществляется контроллером *TableGeneratorController*, с помощью метода *GetGenerateAsync*, в следующем виде (0 – номер пары конца цепочки):

| <i>id</i> красса (Pair)/ <i>id</i> устройства (Device) | № пары (Pair) | <i>id</i> входящей пары (Pair) | № пары в цепочке соединений (вычисление) | <i>id</i> сооружения (Building) | <i>id</i> помещения (Room) |
|--|------------------|-----------------------------------|---|------------------------------------|-------------------------------|
| | | | 0 | | |
| | | | 1 | | |
| | | | 2 | | |

Для выбора рассматриваемой цепочки в визуальной части используется метод *GetListAsync*, выдаваемая таблица

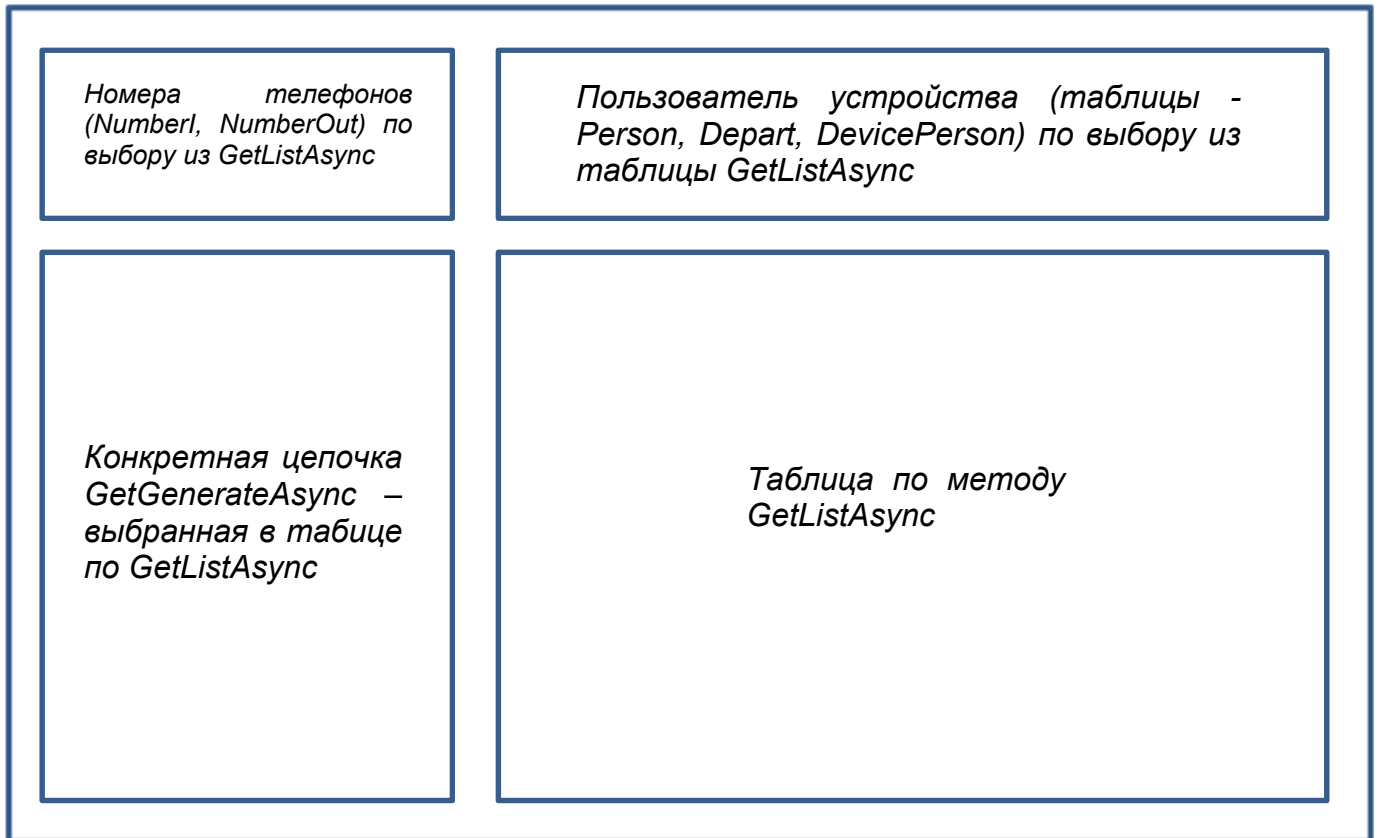
| <i>id</i> красса (Pair)/ <i>id</i> устройств а (Device) начальной пары | № пары (Pair) | <i>id</i> сооруже ния (Building) | <i>id</i> помещени я (Room) | <i>id</i> красса (Pair)/ <i>id</i> устройств а (Device) начальной пары | № пары (Pair) | <i>id</i> сооруже ния (Building) | <i>id</i> помещени я (Room) |
|---|------------------|-------------------------------------|--------------------------------|---|------------------|-------------------------------------|--------------------------------|
| | | | | | | | |

Принцип работы:

Получаем из базы список концов цепочек, затем цикл по концам цепочек с помощью рекурсии находит последовательность соединений от конца к началу присоединяя данные из таблиц *Device*, *Cross*, *Room* – метод *GetGenerateAsync*, получаем первую таблицу, следующий цикл по полученной таблице получает таблицу начала и конца цепочки – метод *GetListAsync*

Экраны визуализации помимо ввода данных:

Главная таблица по результатам обработки:



Редактор соединений

