**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**(Университет ИТМО)**

Факультет **Инфокоммуникационных технологий**

Образовательная программа **Мобильные и облачные технологии**

Направление подготовки (специальность) **09.03.03 Прикладная информатика**

**К У Р С О В О Й П Р О Е К Т**

**по дисциплине «Инфокоммуникационные системы и технологии»**

на тему: «Система агрегации медицинских услуг»

Обучающийся Крылов Михаил Максимович, группа К3140

**Работа сдана**

Дата 07.01.2024

Санкт-Петербург 2023

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_gjdgxs)

[1 Содержание проекта 5](#_30j0zll)

[2 Процессы работы над проектом 6](#_1fob9te)

[3 Задачи, которые стояли передо мной 8](#_3znysh7)

[4 Результаты работы и анализ 11](#_2et92p0)

[5 Оценка работы 12](#_tyjcwt)

[ЗАКЛЮЧЕНИЕ 13](#_x3f8311l5m2k)

**ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

В настоящем отчете о НИР применяют следующие сокращения и обозначения:

1. Парсер – Это программа для сбора и систематизации информации, размещенной на различных сайтах. Источником данных может служить текстовое наполнение, HTML-код сайта, заголовки, пункты меню, базы данных и другие элементы.
2. База данных – это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляется системой управления базами данных (СУБД). Данные вместе с СУБД, а также приложения, которые с ними связаны, называются системой баз данных, или, для краткости, просто базой данных.
3. ETL – (Extract, Transform, Load) – Процесс преобразования сырых данных в обработанные. Включает в себя выгрузку данных из STG слоя, обработку, загрузку обработанных данных в DDS слой.
4. STG (Staging) – слой базы данных, в котором хранятся сырые данные.
5. DDS (Data Detail Store) – слой базы данных, в котором хранятся данные, прошедшую ETL обработку.
6. Docker – это программная платформа для быстрой разработки, тестирования и развертывания приложений.

# **ВВЕДЕНИЕ**

В настоящее время существует огромное количество медицинских клиник, предлагающих различные услуги, и спрос на них постоянно возрастает в связи с растущей потребностью в медицинской помощи. Для того чтобы выбрать подходящее медицинское учреждение из разнообразных вариантов, люди могут воспользоваться специальным агрегатором медицинских услуг.

Наша основная цель заключается в создании уникального агрегатора медицинских услуг, который объединит различных участников медицинской отрасли. Особенностью данного проекта является возможность отслеживания и визуализации изменений в ценах на предоставляемые услуги. Мы предполагаем, что наш агрегатор будет использоваться в качестве подсистемы для системы потенциальных заказчиков. Наша система объективно и безошибочно предоставляет информацию о текущих ценах и их истории. В особенности, наш проект будет очень полезен для многих жителей Санкт-Петербурга, поскольку поможет избавить их от лишних хлопот при поиске выгодного и качественного лечения, которое так важно для здоровья.

# 1 Содержание проекта

Проект “Система агрегации медицинских услуг” заключается в разработке программы, в результате работы которой выводится список медицинских услуг, их цена в определенные промежутки времени и наименование клиник, которые предоставляют данные услуги.

Задачи проекта включают:

1. Установка необходимых инструментов и компонентов на персональные ЭВМ.
2. Продумывание и конструирование базы данных. В соответствии с типом данных, расположенных на сайтах, должны быть обозначенные типы данных в соответствующих колонках базы данных.
3. Написание парсеров и интеграция полученных данных в базу данных в STG слой.
4. Написание ETL скрипта, который будет брать сырые данные из STG слоя, обрабатывать их и вставлять в DDS слой.
5. Развертывание базы данных, парсеров и ETL скрипта в Docker контейнере
6. Передача данных в Yandex DataLens с целью их последующий визуализации

В конце реализации мы получими готовый продукт, выполняющий роль агрегатора медицинских услуг с интерактивным дашбордом. Помимо этого, он обладает быстрой разворачиваемостью за счет Docker-compose и всегда актуальными данными: парсеры работают безостановочно, всё время пополняя базу данных.

# 2 Процессы работы над проектом

В начале работы над проектом наша команда разработала следующий план работы над проектом:

1. Установка необходимых инструментов и компонентов на персональные ЭВМ.:

Данный этап включает в себя установку на персональные компьютеры Python, также среду программирования - PyCharm

1. Проектирование базы данных:

База данных должна содержать два слоя: STG и DDS. В каждом слое должно быть две таблицы: **analyzes** и **addresses**. Таблица **analyzes** включает в себя следующие столбцы:

* Код
* Группа
* Наименование
* Стоимость услуги
* Наименование лаборатории
* Дата

Таблица **addresses** включает в себя следующие столбцы:

* Город
* Адрес
* Контактные номера
* Часы работы
* Станция метро
* Наименование клиники

Также, база данных у нас будет запускаться в контейнере, в связи с чем, следует предусмотреть и разработать скрипт, отвечающий за создание слоев, а также таблиц с определенными столбцами и типами данных.

1. Написание парсеров.

Данный этап подразумевает написание парсеров, закрепленных за участниками проекта. На одного участника проекта один парсер

1. Написание ETL - скрипта.

В данном этапе полученные данные обрабатываются и записываются в новый слой

1. Создание главного файла, который будет запускать поочередно все парсеры. После завершения работы которых он будет запускать ETL скрипт
2. Выгрузка данных.

Данный этап предполагает выгрузку обработанных данных из DDS слоя базы данных и загрузку их в Yandex DataLens для визуализации

1. Контейнеризация

Данный этап подразумевает загрузку всех парсеров в Docker контейнер. Также, запуск базы данных и ETL скрипта в рамках контейнера.

# 3 Задачи, которые стояли передо мной

Первоначально, передо мной стояла задача по установке необходимых компонентов, а именно: язык программирования Python и среда разработки PyCharm. После установки необходимых компонентов, моей задачей было разработать парсер данных сайта <https://www.cmd-online.ru>. Для того, чтобы разработать хороший парсер, сначала нужно проанализировать сайт: удостовериться, что после загрузки сайт не загружает информацию с помощью JavaScript. В ином случае, придется использовать selenium, который сильно нагружает систему, а также имеет низкую производительность. После того, как я удостоверился в том, что сайт не использует JavaScript для нужных нам данных, я приступил к написанию парсера с помощью эффективных библиотек - requests для запросов, BeautifulSoup для удобной работы с html - разметкой. После успешного написания парсера, поговорив с руководителем команды - Гарифуллином Марато Рустамовичом, было принято решения передать задачи по контейнерезации и создании базы данных мне.

Для работы с базой данных мной был реализован модуль, который при иницилизации запускает функции проверки корректности базы данных, корректности слоев. Если проверка провалилась - запускается функция создания слоев и таблиц. Для этого, я использовал асинхронную библиотеку **asyncpg**, которая позволяет в асинхронном порядке отправлять запросы к базе данных. Это обеспечивает высокую скорость добавление новых данных и получения старых. Помимо этого, я реализовал две асинхронные функции, для получения и вставки данных. Асинхронность позволяет сэкономить драгоценное время. Хоть я и использовал асинхронные функции, я постарался их сделать наиболее удобными и простыми, чтобы остальные участники команды могли быстро разобраться как их использовать и интегрировать в свой код. Пример части кода, отвечающую за функции добавления и получения данных из базы данных на рисунке 1.

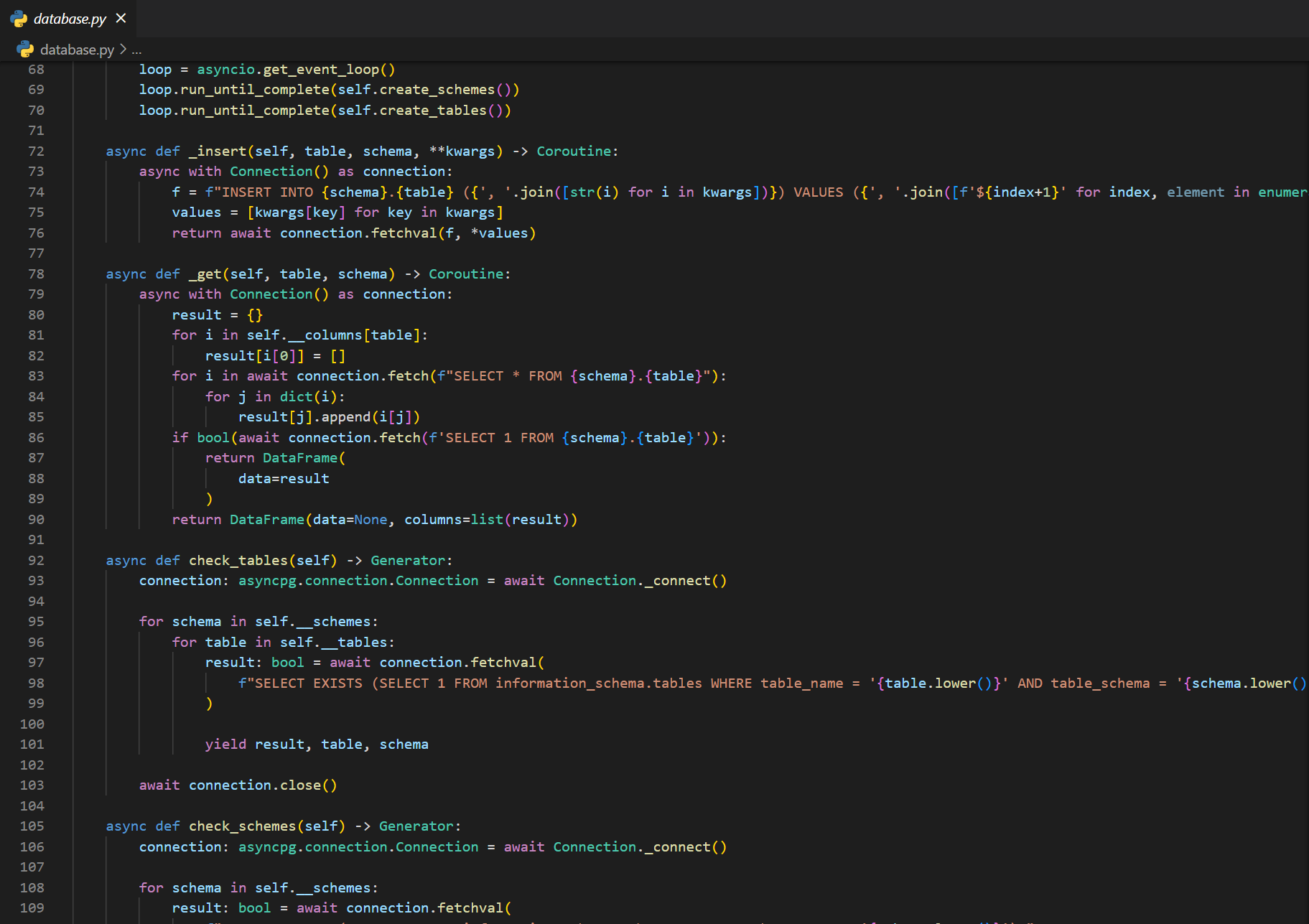


рисунок 1

Для контейнеризации мы использовали **Docker-compose**, позволяющий одновременно запускать несколько **Docker** контейнером и настраивать между ними связь. Мной с нуля было изучена структура работы **Docker** контейнером внутри **Docker-compose**. Помимо этого, мне пришлось исправлять ошибку связи между контейнерами - база данных занимала тот же порт, что занимал **Docker** контейнер, в связи с чем, соединение было невозможно. Решением было добавить в код, связанный с базой данных, проверку на операционную систему, и в зависимости от того, какая операционная система используется - использовать определенный порт. Конечная версия Docker-compose на рисунке 2.

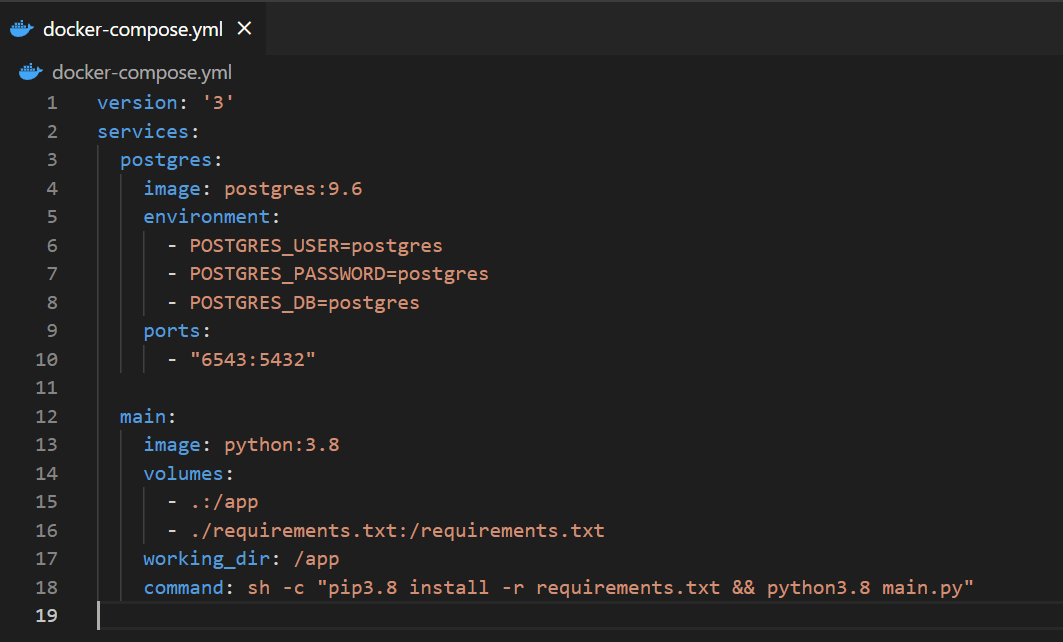


рисунок 2

# 4 Результаты работы и анализ

В ходе разработки нашей команды был создан репозиторий на **github**, который хранит в себе готовый проект по агрегации медицинских данных среди Санкт-Петербургских центров предоставления медицинской помощи. Проект запускается внутри **Docker** контейнера, что делает его удобным и легко развертываемым. База данных быстро и легко восстанавливается, в случае непредвиденных ситуаций. А визуализация в удобном формате помогает понять пользователю самую выгодную клинику и узнать, в какое время повышаются цены на те или иные медицинские услуги.

Лично я впервые столкнулся с **Docker** и **Docker-compose**, это был познавательный и крайне ценный опыт работы. Помимо этого, процесс написания парсеров был увлекателен, а вместе с тем, ценным: на рынке в данный момент недостаток специалистов, занимающихся написанием программ для получения данных.

В будущем планируется использовать другой BI сервис, потому что, в связи с сжатыми сроками, мы использовали Yandex Datalens, потому что он прост в изучении и быстр в освоении. Однако, в этом же и его минусы: отсутствие кастомизации пользовательского интерфейса. Каждый из нас получил ценный опыт как в разработке, так и в командной работе.

# 5 Оценка работы

Реализация агрегатора медицинских услуг получилась лишь потому, что мы работали в команде: каждый из нас в чем-то своем хорош, однако, если бы не командная работа, мы бы не справились. Во время разработки возникали трудности, например: добавления данных в базу данных после парсинга. Для разрешения проблемы мы собирались командой и решали всё вместе, с помощью таких средств, как: Discord, Telegram, Zoom. Все эти приложения помогают лучше коммуницировать участникам команды между собой.

Отдельное слово благодарности стоит выразить нашему руководителю команды: Гарифуллину Марату Рустамовичу, ведь именно он отвечал за раздачу ролей в команде, следил за выполнением заданий и в случае неудаче всегда помогал справиться с проблемой. Также, он был инициатором большой части созвонов. Не оставил нас на произвол судьбы и во время выступления: присутствовал при написании текста к выступлению, а также проводил пробное прослушивание, чтобы проверить нашу готовность. А после дал аргументированную обратную связь и помог нам устранить некоторые проблемы, которые возникли при пробном выступлении. Я считаю, что хоть Гарифуллин Марат Рустамович напрямую не занимался разработкой парсеров и других скриптов, он заслуживает похвалу, ведь именно он следил за нашей работой и корректировал нашу деятельность.

# ЗАКЛЮЧЕНИЕ

В рамках данной научно-исследовательской работы был разработан агрегатор медицинских услуг по городу Санкт-Петербург. Помимо этого, агрегатор обладает возможностью удобной визуализации данных и просмотра истории изменении цен. Проект является готовым продуктом, однако, наша команда планирует дорабатывать дальше.

Каждый из нас добился своих поставленных целей и задач, помимо этого, мы обрели прикладные знания в разработке систем по парсингу данных как с статичных сайтов, так и с динамичных, использующих JavaScript по мере своей работы.

Работа в команде была на высшем уровне: мы все сблизились, начали лучше понимать друг друга и старались помогать в проекте. Мы получили чрезвычайно ценный опыт работы в команде.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Конюхов В.Г. База данных. Понятие, значение и роль в современном мире // Системные технологии. 2017. №24. URL: <https://cyberleninka.ru/article/n/baza-dannyh-ponyatie-znachenie-i-rol-v-sovremennom-mire> (дата обращения: 07.01.2024).
2. Абрамова Татьяна Алфиевна Разработка парсинг-системы для получения скрытых ссылок со страниц социальных сетей // Вестник ПензГУ. 2016. №3 (15). URL: <https://cyberleninka.ru/article/n/razrabotka-parsing-sistemy-dlya-polucheniya-skrytyh-ssylok-so-stranits-sotsialnyh-setey> (дата обращения: 07.01.2024).
3. Буренков И.А. ПРИМЕНЕНИЕ ВИРТУАЛЬНЫХ КОНТЕЙНЕРОВ DOCKER ДЛЯ ЗАПУСКА СЕРВИСОВ // Литьё и металлургия. 2022. №3. URL: <https://cyberleninka.ru/article/n/primenenie-virtualnyh-konteynerov-docker-dlya-zapuska-servisov> (дата обращения: 07.01.2024).