**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 Люсин Дмитрий Витальевич, группа К3139

**Работа сдана**

Дата 07.01.2024

Санкт-Петербург 2023

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc155561013)

[ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ 3](#_Toc155561014)

[ВВЕДЕНИЕ 5](#_Toc155561015)

[1 Суть проекта 6](#_Toc155561016)

[2 Задачи всего проекта 7](#_Toc155561017)

[3 Задачи, поставленные передо мной 8](#_Toc155561018)

[4 Работа над поставленными задачами 9](#_Toc155561019)

[5 Анализ работы 10](#_Toc155561020)

[6 Взаимодействие с командой 11](#_Toc155561021)

[7 Взаимодействие с руководителем 12](#_Toc155561022)

[8 Оценка работы руководителя 13](#_Toc155561023)

[ЗАКЛЮЧЕНИЕ 14](#_Toc155561024)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 15](#_Toc155561025)

[ПРИЛОЖЕНИЕ А 16](#_Toc155561026)

# ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

* ETL - это аббревиатура, которая обозначает три основных этапа в обработке данных: извлечение (Extract), преобразование (Transform) и загрузка (Load). ETL процесс является ключевым компонентом в области управления и анализа данных, особенно в сфере хранения и обработки больших объемов данных.
* Парсинг (или веб-парсинг) – это процесс анализа структуры данных с целью извлечения нужной информации с веб-сайтов.
* Дашборд (dashboard) – это инструмент визуализации данных, предназначенный для отображения ключевой информации, метрик или показателей в удобной и легко воспринимаемой форме.
* Агрегация в общем смысле означает сбор, объединение или сжатие данных в целях получения обобщенной информации или статистики.
* Среда разработки (или интегрированная среда разработки, СР) — это программный инструмент, предназначенный для облегчения процесса создания, отладки и тестирования программного обеспечения.
* Jupyter Notebook — это интерактивная среда для выполнения программ на языках программирования, таких как Python, R, и других. Она представляет собой веб-приложение, позволяющее создавать и обмениваться документами, содержащими живой код, уравнения, визуализации и научные выражения.
* СУБД - это сокращение от "система управления базами данных". Это программное обеспечение, предназначенное для создания и управления базами данных. СУБД обеспечивают эффективное хранение, организацию, поиск, обновление и анализ данных.
* Python - это высокоуровневый, интерпретируемый, язык программирования общего назначения.
* Apache Airflow — это открытое программное обеспечение для планирования, мониторинга и управления выполнением рабочих процессов (или потоков данных). Он предоставляет платформу для создания, планирования и мониторинга рабочих процессов, представленных в виде направленного ациклического графа (DAG).
* DataLens — это универсальный облачный инструмент для анализа и визуализации данных.
* DBeaver — это бесплатная программа для работы с СУБД.
* ClickHouse - это колоночная аналитическая СУБД с открытым кодом.
* Скрипт - это последовательность команд или инструкций, написанных на языке программирования, предназначенная для автоматизации выполнения определенных задач.

# ВВЕДЕНИЕ

Концепция системы агрегации медицинских услуг становится все более актуальной в современном мире, где люди сталкиваются с растущей потребностью в доступе к качественным медицинским услугам. Система агрегации медицинских услуг может значительно упростить процесс выбора и получения медицинской помощи, предоставляя пациентам возможность сравнивать различные предложения от различных медицинских лабораторий, получать информацию о ценах, адресах и предоставляемых услугах. Такая система также может помочь улучшить доступность медицинских услуг для людей, проживающих в удаленных районах или имеющих ограниченные возможности по мобильности. Также система агрегации медицинских услуг может быть полезна для самих медицинских учреждений как способ привлечения новых клиентов.

Цель проекта – разработать систему агрегации предложений медицинских компаний по услугам с возможностью просмотра истории цен на услуги, на языке Python при помощи библиотек, поддерживающих парсинг сайтов.

Задачи:

* Установка необходимых инструментов и компонентов на персональные ЭВМ
* Написание парсеров медицинских сайтов
* Локальное объединение парсеров, СУБД и Airflow
* Написание скриптов предобработки данных парсингов, интеграция скриптов в Airflow
* Визуализация подготовленных данных из Clickhouse в DataLens
* Подготовка документации по проекту

# 1 Суть проекта

Суть проекта по разработке системы агрегации медицинских услуг заключается в создании централизованной платформы, которая объединяет информацию о медицинских услугах от различных медицинских лабораторий. Эта система будет предоставлять пользователям возможность искать, сравнивать и выбирать медицинские услуги на основе их цены.

Для создания такой системы проводится написание парсеров, которые собирают необходимые данные, которые в последствии проходят процесс предобработки. Этот процесс позволяет приводить к общему виду сырые данные, собранные с помощью разных парсеров.

Далее готовые данные были подключены к сервису, где полученные таблицы с данными мы преобразовали в распознаваемые DataLens объекты. На основании значений внутри столбцов были созданы различные чарты с простыми агрегациями, повествующие о данных конечному пользователю.

Таким образом, с помощью системы агрегации пользователи смогут получить доступ к информации о месте расположения медицинского учреждения, наличии метро, телефона для записи, расписании приема, ценах на услуги. Кроме того, система будет предоставлять возможность просмотра истории цен в конкретном медицинском центре и находить самую низкую цену на конкретную услугу среди всех медицинских центров. Также разработка такой системы позволит улучшить доступность и прозрачность медицинских услуг для всех пользователей.

# 2 Задачи всего проекта

* Установка необходимых инструментов и компонентов на персональные ЭВМ
* Написание парсеров медицинских сайтов
* Локальное объединение парсеров, СУБД и Airflow
* Написание скриптов предобработки данных парсингов, интеграция скриптов в Airflow
* Визуализация подготовленных данных из Clickhouse в DataLens
* Подготовка документации по проекту

# 3 Задачи, поставленные передо мной

* Установка необходимых инструментов и компонентов на персональные ЭВМ
* Написание парсера медицинского сайта INVITRO
* Написание скриптов предобработки сырых данных из парсеров
* Объединение скриптов предобработки с базой данных

# 4 Работа над поставленными задачами

4.1 Установка необходимых инструментов и компонентов на персональные ЭВМ

Все необходимые инструменты и компоненты, на момент начала работы, уже были установлены на ЭВМ

4.2 Написание парсера медицинского сайта INVITRO

Написание парсера происходило на языке программирования Python. Главными задачами написанной программы были сбор данных двух типов: «Адреса» и «Анализы», а также запись всех собранных данных в файлы типа .csv.

4.3 Написание скриптов предобработки сырых данных из парсеров

Так как каждый член команды работал с отдельным сайтом, то одни и те же данные могли быть по-разному записаны в .csv файлы, поэтому было необходимо написать скрипты предобработки этих сырых данных. В результате работы скрипты приводили к общему виду данные о номерах телефонов, городах и станциях метро в файле «Адреса» и данные о дате запроса и коде анализов в файле «Анализы».

4.4 Объединение скриптов предобработки с базой данных

Сутью задачи было объединить уже существующие скрипты с базой данных путём разработки модуля, который позволял реализовывать запрос на добавление данных в базы данных.

# 5 Анализ работы

В ходе участия в курсовом проекте "Система агрегации медицинских услуг" я получил как теоретические знания о парсинге и базах данных, так и практические – мне довелось написать скрипт для парсинга медицинского сайта. Также благодаря работе над данным проектом я получил очень ценный опыт – опыт работы в команде и взаимодействие с ранее не знакомыми людьми.

В результате всей работы в первую очередь был написан скрипт для парсинга данных с сайта INVITRO. Далее написан алгоритм предобработки сырых данных парсеров для приведения этих самых данных к общему виду для дальнейшей удобной работы с присоединением всех обработанных данных к базе данных. В конце соответственно произошло объединение скрипта предобработки с базой данных.

Первые трудности возникли на этапе написания парсера. Из-за сложного строения сайта для отдельных данных приходилось искать решения для успешной реализации скрипта. Также возникали трудности при выполнении задачи по объединению скриптов предобработки и базы данных.

# 6 Взаимодействие с командой

На этапе написания парсеров члены команды работали обособленно друг от друга, однако уже тогда мы проводили сборы вместе с руководителем в Discord для обсуждения хода работы. После написания парсеров с целью экономии времени все члены команды были поделены на кросс-команды, в результате чего увеличилось взаимодействие между членами команды. Пиком командной работы стал день защиты, когда вся команда обсуждала план защиты. Вплоть до защиты команда регулярно собиралась на обсуждения в Discord.

# 7 Взаимодействие с руководителем

С руководителем я взаимодействовал на протяжении всей работы. Благодаря руководителю мне удалось закончить скрипт парсинга. По моей просьбе он оценил скрипт предобработки данных и проконсультировал по работе с объединением скрипта и базы данных. Также руководитель организовывал сбор всех участников команды в Discord.

# 8 Оценка работы руководителя

Взаимодействие с руководителем проекта происходило дистанционно. Руководитель всегда был готов развёрнуто ответить на интересующие вопросы и помочь в решении трудных моментов при выполнении задач. Во время общения руководитель был очень доброжелателен. ВО время помощи в решении задач была видна опытность руководителя. Я доволен работой руководителя и рад, что мне довелось работать с таким человеком.

# ЗАКЛЮЧЕНИЕ

В общем цель проекта была достигнута – командой была разработана система агрегации медицинских услуг. Однако в ходе работы для того чтобы уложится в дедлайн пришлось поделить команду на кросс-команды для оперативной работы над поставленными задачами. Каждый член команды отметил для себя область, в которой он может продолжить развитие. Также несмотря на достигнутую цель, полученную систему нельзя назвать идеальной, так как имеются ещё немало функций, которые можно внедрить в неё. Например возможность кастомизации пользовательского интерфейса или полная автоматизация процесса.

Моя работа в данном проекте была направленна на написание скрипта парсера и скрипта для предобработки сырых данных. Наиболее важной из двух являлась задача по написанию скрипта предобработки данных, так как на его основе происходила установка связи между скриптом и базой данных.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Основы парсинга на Python: от Request да Selenium [Электронный ресурс] – URL: https://habr.com/ru/companies/selectel/articles/754674/

2 Python: основы скрапинга и парсинга [Электронный ресурс] – URL: https://proglib.io/p/samouchitel-po-python-dlya-nachinayushchih-chast-17-osnovy-skrapinga-i-parsinga-2023-03-13?ysclid=los6qu8c84405447993

3 Предварительная обработка данных с помощью библиотеки Pandas (Задача) [Электронный ресурс] – URL: https://habr.com/ru/articles/727228/

4 Работа с Pandas: ультимативный гайд по библиотеке [Электронный ресурс] – URL: https://tproger.ru/articles/gajd-po-obrabotke-dannyh-s-pomoshhyu-pandas-chast-pervaya?ysclid=lq9b4wgn1k778409299

5 МЕЖГОСУДАРСТВЕННЫЙ СТАНДАРТ. Система стандартов по информации, библиотечному и издательскому делу. ОТЧЕТ О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ. Структура и правила оформления (ГОСТ 7.32–2017) / Стандартинформ - М., 2017. – 32 с. – URL: https://cs.msu.ru/sites/cmc/files/docs/2021-11gost\_7.32-2017.pdf

# ПРИЛОЖЕНИЕ А

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Название проекта

Система агрегации медицинских услуг

Цель

2.1. Цель проекта – разработать систему агрегации предложений медицинских компаний по услугам с возможностью просмотра истории цен на услуги, на языке Python при помощи библиотек, поддерживающих парсинг сайтов за 49 дней.

2.2. Назначение проекта – система агрегации медицинских услуг для пользователей выступит полезным инструментом при поиске конкретных медицинских услуг, предоставляемых во множестве специализированных центров и лабораторий города. Для бизнеса, система агрегации медицинских услуг может быть эффективным инструментом по привлечению новых клиентов.

Сроки выполнения

Начало 01 ноября 2023 г. Конец 20 декабря 2023 г.

Исполнитель проекта (руководитель проекта)

Гарифуллин Марат Рустамович

Термины и сокращения

* ETL - это аббревиатура, которая обозначает три основных этапа в обработке данных: извлечение (Extract), преобразование (Transform) и загрузка (Load). ETL процесс является ключевым компонентом в области управления и анализа данных, особенно в сфере хранения и обработки больших объемов данных.
* Парсинг (или веб-парсинг) – это процесс анализа структуры данных с целью извлечения нужной информации с веб-сайтов.
* Дашборд (dashboard) – это инструмент визуализации данных, предназначенный для отображения ключевой информации, метрик или показателей в удобной и легко воспринимаемой форме.
* Агрегация в общем смысле означает сбор, объединение или сжатие данных в целях получения обобщенной информации или статистики.
* Среда разработки (или интегрированная среда разработки, СР) — это программный инструмент, предназначенный для облегчения процесса создания, отладки и тестирования программного обеспечения.
* Jupyter Notebook — это интерактивная среда для выполнения программ на языках программирования, таких как Python, R, и других. Она представляет собой веб-приложение, позволяющее создавать и обмениваться документами, содержащими живой код, уравнения, визуализации и научные выражения.
* СУБД - это сокращение от "система управления базами данных". Это программное обеспечение, предназначенное для создания и управления базами данных. СУБД обеспечивают эффективное хранение, организацию, поиск, обновление и анализ данных.
* Python - это высокоуровневый, интерпретируемый, язык программирования общего назначения.
* Apache Airflow — это открытое программное обеспечение для планирования, мониторинга и управления выполнением рабочих процессов (или потоков данных). Он предоставляет платформу для создания, планирования и мониторинга рабочих процессов, представленных в виде направленного ациклического графа (DAG).
* DataLens — это универсальный облачный инструмент для анализа и визуализации данных.
* DBeaver — это бесплатная программа для работы с СУБД.
* ClickHouse - это колоночная аналитическая СУБД с открытым кодом.
* Скрипт - это последовательность команд или инструкций, написанных на языке программирования, предназначенная для автоматизации выполнения определенных задач.

Технические требования

6.1. Технические требования к компонентам и инструментам:

* Создание скриптов будет происходить в интегрированной среде разработки PyCharm на языке Python использованием библиотек Beautiful Soup, Scrapy, Requests и других. Расширение файлов скриптов «.py».
* Хранение данных, полученных от работы парсеров будет выполняться за счет колоночной СУБД Clickhouse, взаимодействие с которой будет осуществляться через DBeaver.
* Запуск скриптов для актуализации данных в Clickhouse будет осуществляться по расписанию с помощью Airflow.
* Для подготовки данных к визуализации полученные данные от парсингов будут выступать источником данных и обрабатываться при помощи скриптов, написанных в JupyterHub в формате «.ipynb». Далее конвертируем их в «.py» для запуска в Airflow. Подготовленные данные записываются в отдельные таблица Clickhouse.
* Визуализация данных будет осуществляться в DataLens, источником данных выступят таблицы с подготовленными данными.
* Система управления версиями кода и репозиториев – GitLab.
* Версия основных компонентов: Python (3.10), Airflow (2.7.2), pandas (2.0.3), numpy (1.26.0), clickhouse-driver (0.2.6), requests (2.31.0), beautifulsoup4 (4.12.2), JupyterLab (3.2.6)

6.2 Требования к продукту

* Поиск и фильтрация. Пациенты должны иметь возможность искать медицинские услуги на основе различных критериев. Система должна предоставлять возможности фильтрации результатов поиска.
* Статистика и аналитика. Возможность сбора и анализа статистических данных о стоимости услуг, представленных в виде графиков и диаграмм.
* Данные должны быть актуальны для текущего дня.

6.3 Потребители

Система агрегации медицинских услуг может иметь разных пользователей и потребителей:

* Пациенты являются основными потребителями медицинских услуг. Для них система предоставляет возможность поиска фильтрации медицинских услуг в специализированных центрах и лабораториях.
* Партнеры и поставщики медицинских услуг. Другие медицинские учреждения, лаборатории, другие поставщики услуг могут использовать систему для взаимодействия с другими участниками системы, обмена информацией и предоставления своих услуг.

Содержание работы

7.1. Этапы проекта

Этапы проекта подробнее описаны в таблице 1.

Таблица 1. Этапы проекта

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Этапы проекта | Сроки выполнения этапов | Ответственный за этап | Вид представления результатов этапа |
| 1 | Разработка технического задания | 10 ноября | Гарифуллин М.Р. | Документ формата .docx с подробным описанием технического задания |
| 2. | Установка необходимых инструментов и компонентов на персональные ЭВМ. | 13 ноября | Все участники проекта. Контролирующий - Тихонова Марина Сергеевна | Устное заявление контролирующему. |
| 3. | Написание парсеров медицинских сайтов. | 27 ноября | Все участники проекта. Контролирующий – Тихонова Марина Сергеевна.  Code review: Гарифуллин М.Р. | Выгрузка .py файла со скриптом в репозиторий. |
| 4. | Создание Docker контейнера с Airflow и СУБД. | 27 ноября | Гарифуллин М.Р. | Docker file. |
| 5. | Локальное объединение парсеров, СУБД и Airflow | 3 декабря | Все участники проекта. Контролирующий – Тихонова Марина Сергеевна | Запуск скриптов по расписанию, запись данных в БД. |
| 6. | Написание скриптов предобработки данных парсингов, интеграция скриптов в Airflow. | 9 декабря | Все участники проекта. Контролирующий – Тихонова Марина Сергеевна | Выгрузка обработанных данных в таблицы Clickhouse. |
| 7. | Визуализация подготовленных данных из Clickhouse в DataLens | 15 декабря | Все участники проекта. Контролирующий – Тихонова Марина Сергеевна | Интерактивные дашборды по подготовленным данным. |
| 8. | Составление отчета и подготовка к выступлению | 19 декабря | Все участники проекта. Контролирующий – Тихонова Марина Сергеевна | Индивидуальные отчеты и общая презентация. |
| 9. | Защита проекта (сдача отчета и представление доклада с презентацией) | 20 декабря |  | Оценка в ведомости. |

7.2. Индивидуальное распределение ролей и задач.

Процесс разработки системы агрегации медицинских услуг разделен на последовательные этапы, поэтому для большего погружения участников проекта в новые технологии каждый из участников примет на себе несколько ролей. Распределение ролей представлено в таблице 2.

Таблица 2. Индивидуальное распределение ролей и задач проекта

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | ФИО участника | Роли в проекте | Задачи к выполнению | Индивидуальное дополнение |
| 1. | Гарифуллин Марат Рустамович | Tech Lead | 1, 4. |  |
| 2. | Крылов Михаил Максимович | Python developer, BI developer | 2,3,5,6,7,8,9 | Сайт для парсинга: https://www.cmd-online.ru |
| 3. | Люсин Дмитрий Витальевич | Python developer, BI developer | 2,3,5,6,7,8,9 | Сайт для парсинга: https://www.invitro.ru |
| 4. | Рыжков Никита Владимирович | Python developer, BI developer | 2,3,5,6,7,8,9 | Сайт для парсинга: https://gemotest.ru |
| 5. | Тихонова Марина Сергеевна | Python developer, BI developer, business analyst | 2,3,5,6,7,8,9 | Сайт для парсинга: https://lab4u.ru |
| 6. | Шевченко Семён Олегович | Python developer, BI developer | 2,3,5,6,7,8,9 | Сайт для парсинга: https://riat.spb.ru |

Основные результаты работы и формы их представления

* Пример действующего скрипта в airflow на рисунке 1.

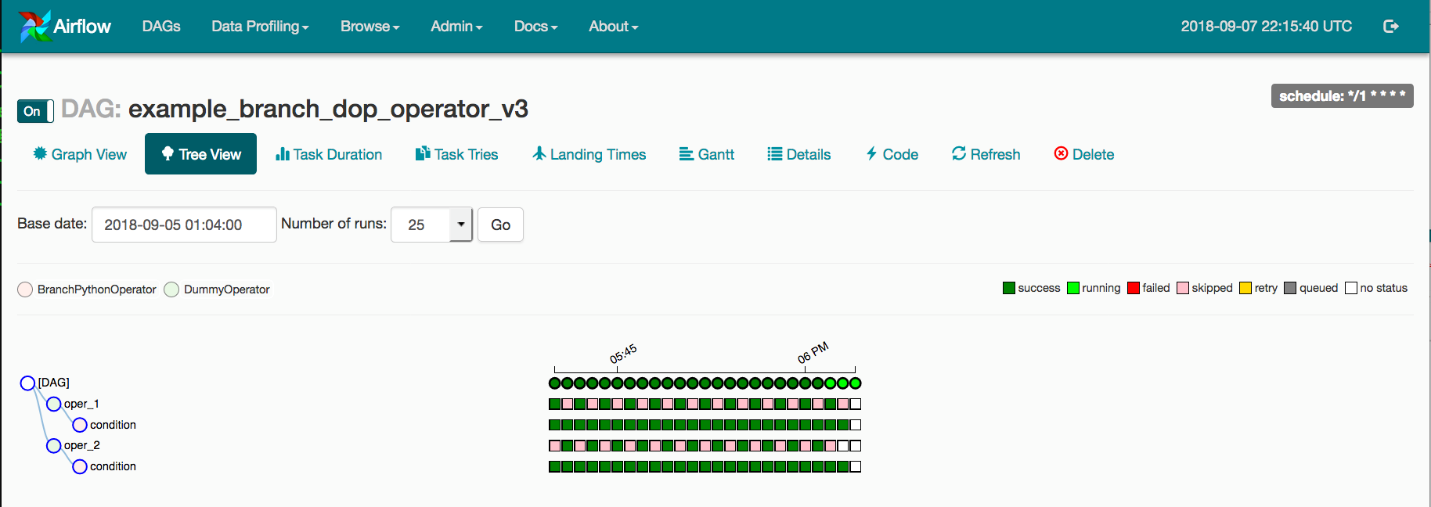


Рисунок 1 - Скрипт в airflow

* Пример дашборда DataLens на рисунке 2.

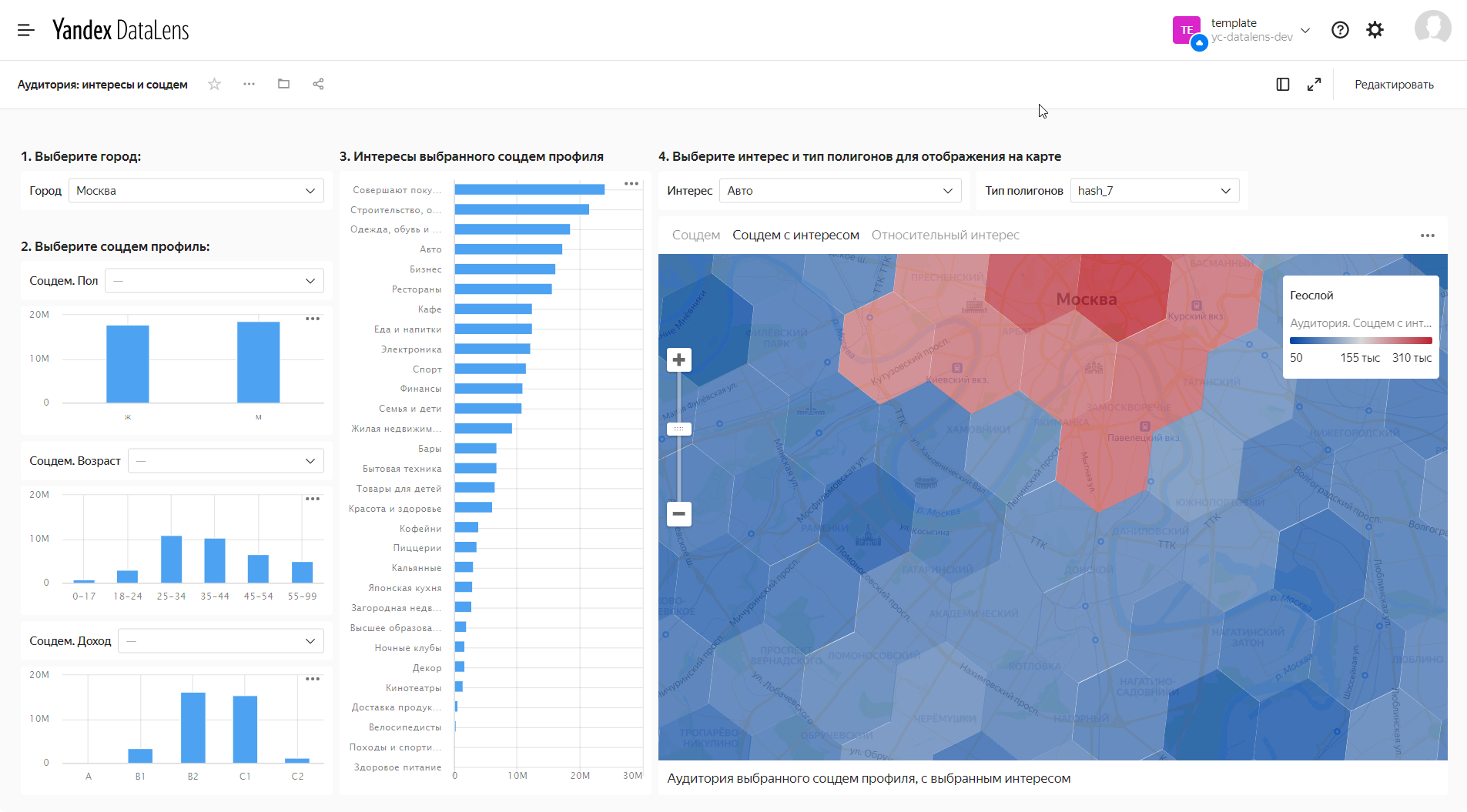


Рисунок 2 - Дашборд DataLens