Санкт-Петербургский национальный исследовательский
университет информационных технологий, механики и оптики

Факультет прикладной информатики
Направление подготовки 11.04.02

Домашнее задание

Выполнил:
Швалов Даниил Андреевич K4112c
Проверила:
Болгова Екатерина Владимировна

Санкт-Петербург
2025

СОДЕРЖАНИЕ

1 Введение

Цель работы: разработать уточненную диаграмму классов.

2 Ход работы

Для ГИС для поиска точек общественного питания была разработанная уточненная диаграмма классов, представленная на рисунке 1.

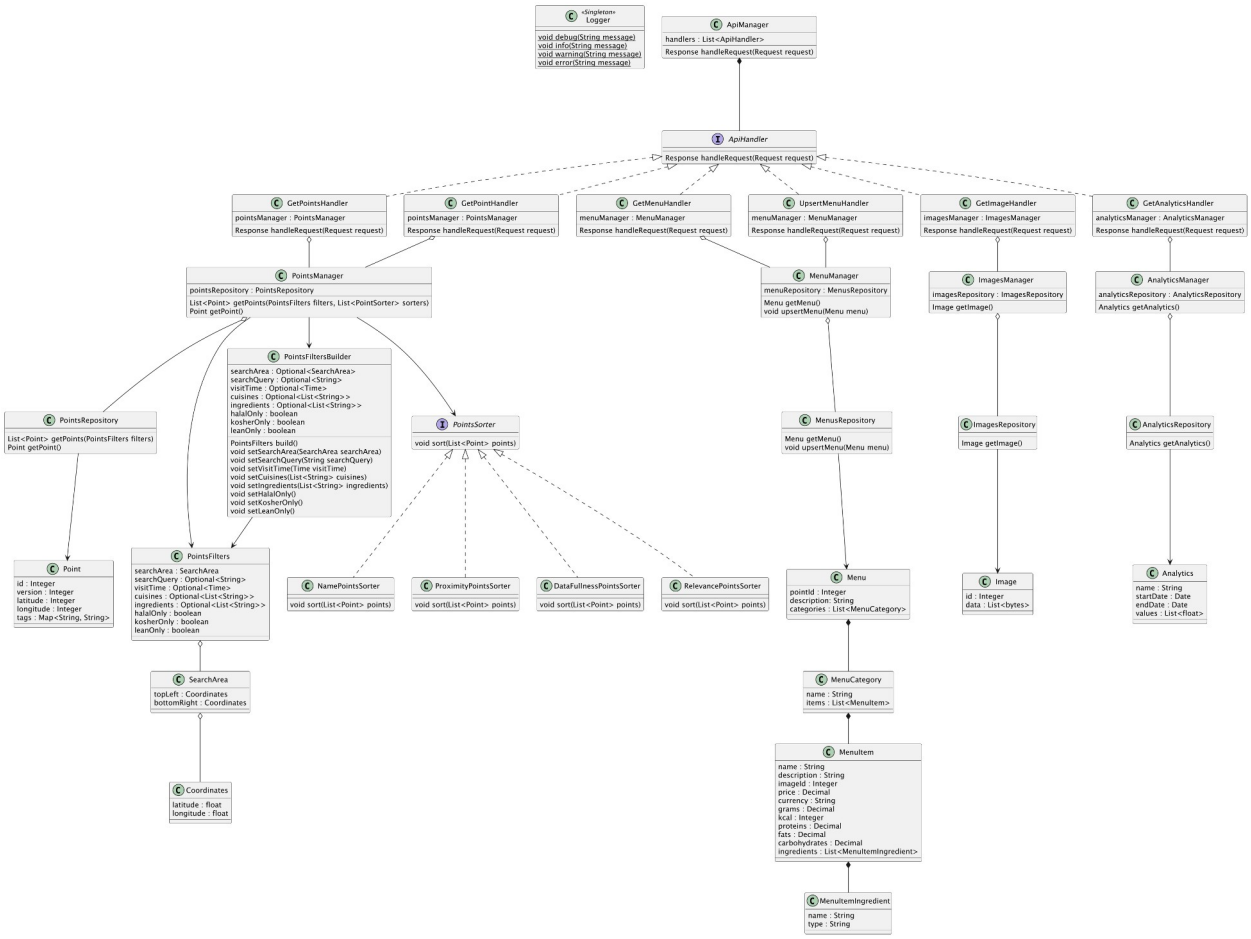

Рисунок 1 — Уточненная диаграмма классов

На рисунке 1 представлены следующие классы:

На ней представлены все основные классы, присутствующие в системе:

1) Point — это класс, который представляет информацию о точке (например, расположение, теги и т. п.);

2) MenuItemIngredient — это класс, который содержит информацию об ингредиентах блюд (например, название, тип);

3) MenuItem — это класс, который содержит информацию о позиции в

меню (например, название, стоимость, вес и т. п.). Включает в себя MenuItemIntgredient;

4) MenuCategory — это класс, который содержит информацию о категориях в меню. Включает в себя MenuItem;

5) Menu — это класс, который представляет информацию о меню. Включает в себя MenuCategory;

6) Image — это класс, который представляет из себя информацию о изображении (например, содержимое изображения в байтах, мета информация);

7) Analytics — это класс, который представляет из себя аналитическую информацию (например, название, значения);

8) PointsRepository, MenuRepository, ImagesRepository, AnatylicsRepository — это классы, который предоставляют методы для доступа к базе данных, а также преобразуют данные из базы данных в соответствующие классы;

9) PointsSorter — это интерфейс, который используется для реализации конкретных алгоритмов сортировки точек;

10) NamePointsSorter, ProximityPointsSorter, DataFullnessPointsSorter, RelevancePointsSorter — это классы, которые реализуют конкретные алгоритмы сортировок точек;

11) PointsFilters — это класс, который хранит информацию о параметрах сортировки точек;

12) PointsFiltersBuilder — это класс, который предоставляет удобный интерфейс для создания экземпляров класса PointsFilters;

13) PointsManager, MenuManager, ImagesManager, AnalyticsManager — это классы, которые реализуют бизнес-логику операций доступа и изменения к объектам (например, к меню или точкам). Данные классы используют соответствующие классы-репозитории для доступа к данным;

14) ApiHandler — это интерфейс, который используется для реализации

конкретных обработчиков;

15) GetPointsHandler, GetPointHandler, GetMenuHandler, UpsertMenuHandler, GetImageHandler, GetAnalyticsHandler — это классы, которые реализуют обработку запросов и передают данные в понятном виде менеджерам;

16) ApiManager — это класс, который обрабатывает все входные запросы и передает их соответствующим обработчикам.

При проектировании классов были применены следующие шаблоны GoF:

1) **фасад.** В качестве фасада выступает ApiManager, т. к. он предоставляет единую точку входа в систему в виде функции handleRequest, которая принимает и возвращает универсальные форматы запросов и ответов. Этот шаблон был применен потому, что он позволяет упростить взаимодействие с системой, снизить связность и улучшить читаемость кода;

2) цепочка обязанностей. В качестве цепочки обязанностей выступает интерфейс ApiHandler и все реализации данного интерфейса. ApiHandler предоставляет единый интерфейс обработки запросов, благодаря чему в ApiManager есть возможность построить цепочку проверки соответствия запроса и обработчика. Этот шаблон позволяет упростить добавление новых обработчиков, снижает связность и улучшает масштабируемость системы;

3) строитель. В качестве строителя выступает класс PointsFiltersBuilder. Он позволяет простым образом составить набор конкретных фильтров без необходимости заполнять значения для всех фильтров сразу. Этот шаблон позволяет упростить код, пошагово создавать сложные объекты, изолировать процесс создания объекта от его представления;

4) стратегия. В качестве стратегии выступает интерфейс PointsSorter и все реализации данного интерфейса. Он предоставляет интерфейс для сортировки точек по различным критериям. Благодаря использованию

интерфейса появляется возможность добавлять новые типы сортировок без необходимости изменять существующий код. Этот шаблон позволяет инкапсулировать внутреннюю логику, увеличить гибкость, повысить переиспользуемость, улучшить читаемость;

5) одиночка. В качестве одиночки выступает класс Logger, который предоставляет методы для текстового журналирования информации в стандартный вывод. Класс Logger позволяет упростить работу с журналированием, поскольку для журналирования нет необходимости каждый раз создавать отдельный экземпляр класса. Этот шаблон позволяет получить глобальную доступность, контроль над ресурсами и упростить архитектуру.

3 Вывод

В ходе выполнения данной лабораторной работы была разработана уточненная диаграмма классов.