Управление Цифровизации города Алматы

Акционерное общество «Центр развития города Алматы»

Алматы қаласы цифрландыру басқармасы Арқандібай А. Ә.

# ОТЧЕТ

## по предоставлению данных системы мониторинга индикаторов развития города Алматы (Ситуационный центр) за период с 01.10.2025 по 31.12.2025

(договор №21 от 26 мая 2025 г.)

Заместитель Председателя Правления

АО «Центр развития города Алматы» _____ А.Т. Жексенбай

подпись

Управляющий директор

АО «Центр развития города Алматы» _____ Р.И. Бармашёв

подпись

**г. Алматы, 2025**

СПИСОК ИСПОЛНИТЕЛЕЙ

по пунктам Технической спецификации 3.1. «Развитие Системы мониторинга» и 3.2. «Доступ к системе мониторинга и анализ социально-экономического развития города Алматы»

Директор Ситуационного центра

АО «Центр развития города Алматы» _____ М. Цай

подпись

по пункту 3.3. Технической спецификации

«Обеспечение хранения данных»

Директор Департамента ИТ поддержки

АО «Центр развития города Алматы» _____ И. Беликов

подпись

Алматы қаласы цифрландыру басқармасы\Адильбай А. Ә.

# Оглавление

1. Общие сведения ..... 10
2. Сопровождение модулей Ситуационного центра ..... 12
2.1. Инвентаризация. Модуль «Инфраструктура города» ..... 12
2.1.1. Цены на недвижимость ..... 12
2.1.2. Промышленные объекты ..... 14
2.1.3. Территориальное распределение по школам ..... 15
2.1.4. Территориальное распределение по поликлиникам ..... 16
2.2. Разработка мобильного приложения ..... 17
2.2.1. Разработка сервиса цифровой инклюзии ..... 17
2.2.2. Разработка административной панели для обратной связи от пользователей ..... 34
2.2.3. Тестирование пользовательских сценариев ..... 37
2.2.4. Публикация приложения в Google Play и App Store ..... 38
2.3. Реализация модуля «Мониторинг районов». ..... 39
2.3.1. Развитие администрации данных под дополнительные параметры ..... 39
2.3.2. Разработка Карты для демонстрации информации по строительным работам… ..... 40
2.4. Разработка модуля «Мониторинг сейсмоустойчивости городских объектов» ..... 53
2.4.1. Разработка системы уведомлений для мониторинга изменений в статусе объектов ..... 53
2.4.2. Рекомендации по сейсмоустойчивости для усиления уязвимых объектов ..... 55
2.5. Разработка модуля «Мониторинг сейсмической опасности» ..... 62
2.5.1. Настройка системы формирования ежедневной отчетности о сейсмических событиях ..... 64
2.6. Разработка модуля «Мониторинг договоров» ..... 65
2.6.1. Добавление в форму регистрации данных функционала по регистрации договоров на разработку ПСД ..... 65
2.6.2. Реализация мониторинга и анализа данных по договорам на строительство с выявлением аффилированных компаний ..... 68
2.6.3. Аналитика по заказчикам ..... 72
2.6.4. Реализация мониторинга и анализа данных по договорам ПСД ..... 76
2.7. Разработка сервиса «Карта туристических объектов города Алматы» ..... 77
2.7.1. Внедрение функции формирования отчетов по параметрам, включая популярность объектов, сезонность посещений и предпочтения туристов ..... 78
2.7.2. Тестирование и запуск системы для пользователей ..... 79
2.8. Разработка сервиса «Цифровой рейтинг школ». ..... 81

Алматы қаласы цифрландыру басқармасы/Дәнібай А. Э.

2.8.1. Проведение пилотного запуска сервиса на стороне Заказчик ..... 81
2.8.2. Обучение пользователей работе с системой, с предоставлением обучающих материалов, и сбор обратной связи ..... 83
2.9. Разработка модуля «Мониторинг госпитализаций и коечной мощности»…. ..... 84
2.9.1. Анализ данных ..... 85
2.9.2. Разработка рекомендаций по улучшению управления коечной мощностью ..... 89
2.10. Разработка модуля «Карта нозологий и смертности» ..... 92
2.10.1. Реализовать обновления данных на постоянной основе посредствам интеграции ..... 93
2.10.2. Визуализация данных на карте города с детализацией по участкам ВОП…… ..... 93
2.10.3. Разработка визуализации данных, превышающих городские и Республиканские показатели ..... 99
2.11. Развитие модуля «Цифровая инклюзия» ..... 101
2.11.1. Разработка функционала по регистрации факта адаптации объекта. 101
2.12. Развитие модуля «Аналитика Open Almaty» ..... 102
2.12.1. Разработка рейтинга Исполнительных органов с помощью LLM обработки обращений, ответов, комментариев. ..... 102
2.12.2. Реализация модуля по согласованному макету ..... 106
2.12.3. Государственная программа «Taza Qazaqstan» ..... 108
2.13. Развитие модуля «Аналитика нагрузки дорог» ..... 110
2.13.1. Формирование индекса состояния дороги (PCI), классов и рекомендаций ..... 111
2.13.2. Разработка усовершенствованных алгоритмов предикативной аналитики износа дорог ..... 114
2.13.3. Создание интерфейса для визуализации зон повышенного износа на основе новых данных ..... 117
2.14. Развитие чат-бота «Almaty Smart Bot (ASB)» ..... 120
2.14.1. Внедрение ChatGPT API или другой ИИ-модели - замена локальной модели + повышение производительности ..... 121
2.14.2. Добавление дополнительных таблиц до 10 шт ..... 122
2.14.3. Добавление функционала по хранению истории переписки ..... 123
2.15. Разработка модуля «Прогноз дефицита мест в школах» ..... 123
2.15.1. Разработка форм регистрации и редактирования данных ..... 124
2.15.2. Реализация отображения результатов расчета на карте ..... 137
2.16. Анализ парков и скверов ..... 141
2.16.1. Визуализация территориального распределения парков и скверов с учетом поли-центричности города ..... 142
2.16.2. Оценка доступности значимых зеленых зон для различных слоев населения ..... 144
2.16.3. Расчет обеспеченности населения зелеными зонами на основе

Алматы қаласы цифрландыру басқармасының Әлеңдбай Ә.

международных стандартов и демографических данных ..... 145
2.16.4. Разработка рекомендаций по размещению новых парков ..... 147
2.16.5. Разработка интерактивного карты с поддержкой динамического применения слоев данных влияющих на результат ..... 147
2.17. Развитие инфраструктуры для отдыха и спорта ..... 151
2.17.1. Оценка потребностей новых спортивных зон и площадок ..... 153
2.17.2. Рекомендации для реализация пилотных проектов по созданию спортивной инфраструктуры ..... 159
2.17.3. Разработка интерактивного карты с поддержкой динамического применения слоев данных влияющих на результат ..... 164
2.18. Модуль «Мониторинг наркозависимых для Прокуратуры» ..... 166
2.18.1. Обработка данных со следующих ИС (при наличии данных у заказчика): ИС «Регистр наркологических больных», ИС «Спектрум», ЕРАП, ЕРДР…… ..... 167
2.18.2. Формирование архитектуры с принципом «privacy by design» ..... 167
2.18.3. Разработка панели администрирования пользователей ..... 168
2.18.4. Настройка ограничений доступа к данным с логированием действий пользователей ..... 169
2.18.5. Разработка макета модуля ..... 172
2.18.6. Настройка работы модуля через ЕТС ГО ..... 175
2.19. Сопровождение «ЕХД». ..... 175
2.19.1. Сопровождение модуля Единого хранилища данных ..... 176
2.19.2. Интеграционное взаимодействие с ТОО «Центр информационных систем» по проекту АИС «Население» ..... 176
2.19.3. Отработка запросов с текущим пользователем – «Алсеко», и с потенциальными заказчиками в дальнейшем (по согласованию Заказчика) ... 178
2.20. Сопровождение модуля ЕСВМ ..... 178
2.20.1. Визуализацию на ГИС карте камер в рамках проекта «Единая система видео-мониторинга» (ЕСВМ), размещенных в городской среде ..... 178
2.20.2. Обновление данных по местоположению камер ..... 180
2.20.3. Сопровождение и поддержка взаимодействия СЦ с ЕСВМ ..... 181
2.20.4. Предоставление аналитической информации и рекомендаций по установке камер видеонаблюдения на ежемесячной основе. ..... 182
2.20.5. Структурирование и категоризация видеокамер, загрузка данных по работоспособности камер ..... 184
2.20.6. Обработка статистической отчётности. ..... 184
2.20.7. Технологический контроль и мониторинг функциональности и работоспособности функциональных элементов по проектам «Обеспечение комплексной системы безопасности» ..... 186
2.21. Интеграция с Damubala и Bilimalmaty. ..... 189
2.21.1. Разработка алгоритмов для сравнения и идентификации совпадений среди зарегистрированных детей в обеих системах ..... 189
2.21.2. Внедрение инструментов анализа для выявления "мертвых душ" и

Алматы, қаласы, инфрландыру басқармасы\Адәнлібай А. Ә.

случаев дублирования ... 193
3. Доступ к Системе мониторинга ... 196
3.1. Модули с открытым доступом для жителей ... 196
3.1.1. Лендинг ... 196
3.1.2. ГИС «Инфраструктура города» ... 198
3.1.3. Сервис «Мониторинг районов» (Ремонтно-строительные работы) ... 199
3.1.4. Модуль «Бюджет народного участия» ... 200
3.1.5. Модуль «ЦУР» ... 201
3.1.6. Модуль «Индикаторы районов» ... 203
3.1.7. Модуль «Показатели СЭР» ... 204
3.1.8. Модуль «СЗПТ» ... 205
3.1.9. Модуль «Цифровая инклюзия» ... 206
3.1.10. Модуль «Мониторинг образования» ... 209
3.1.11. Модуль «Кружки и секции» ... 210
3.1.12. Модель «Экологический мониторинг ... 211
3.1.13. Модуль «Зеленые насаждения» ... 214
3.1.14. Модуль «Мониторинг транспорта» ... 215
3.1.15. Сервис предложения инициатив ... 215
3.1.16. Модуль «Маслихат» ... 216
3.1.17. Сервис «Умная благотворительность» ... 217
3.1.18. Сервис «Фудшеринг» ... 217
3.1.19. Модуль «Проекты ИКТ» ... 218
3.2. Модули с закрытым доступом ... 220
3.2.1. ГИС «Инфраструктура города. Коммуникации» ... 220
3.2.2. ГИС «Городские видеокамеры» ... 220
3.2.3. Шаговая доступность соц. объектов ... 225
3.2.4. Рекомендация по размещению соц. объектов ... 225
3.2.5. Социально-экономический портрет города ... 226
3.2.6. Проектный офис ... 232
3.2.7. Анализ занятости и доходов населения ... 243
3.2.8. Аналитика дохода населения ... 243
3.2.9. Аналитика аренды ... 245
3.2.10. Анализ вакансий ... 247
3.2.11. Модуль «ДЧС» ... 249
3.2.12. Аналитический модуль «Аналитика ДЧС» ... 250
3.2.13. Модуль «Мониторинг ЧС» ... 253
3.2.14. Рекомендации по пожарным депо ... 254
3.2.15. Анализ и рекомендации кнопок SOS ... 258
3.2.16. Аналитика участковых пунктов ... 262
3.2.17. Анализ дорожных нарушений ... 265
3.2.18. Рекомендации по установке датчиков ... 267
3.2.19. Транспортная доступность общественных пространств ... 269
3.2.20. Анализ общественного транспорта ... 270

Алматы қаласы инфрақұрылым басқармасы\Әдәндібай А. Ә.

3.2.21. Аналитика нагрузки дорог ..... 276
3.2.22. Модуль «Встречи акима с населением» ..... 280
3.2.23. Модуль «Оценка цифровизации» ..... 282
3.2.24. Административный центр ..... 283
4. Обеспечение эффективности хранения данных ..... 284
4.1. Цели и задачи ..... 284
4.2. Системы серверного пространства ..... 284
4.3. Спецификация размещенного оборудования Заказчика ..... 291
4.4. Услуги предоставления доступа к каналам связи ..... 296
4.5. Управление событиями ..... 296
4.6. Управление инцидентами ..... 298
4.7. Доступ к облачному хранилищу Nextcloud ..... 299

Алматы қаласы цифрландыру басқармасы\АДәндібай А. Ә.

# СПИСОК СОКРАЩЕНИЙ, ТЕРМИНОВ И ОПРЕДЕЛЕНИЙ

| Термин | Определение |
|---|---|
| АО ЦРА | Акционерное общество «Центр развития города Алматы» |
| АПК | Аппаратно-программный комплекс |
| АСГПТ | Системы автоматического газового пожаротушения |
| АСУДД | Автоматизированная система управления дорожным движением |
| БД | База данных |
| ВРП | Валовой региональный продукт |
| ГИС | Геоинформационная система |
| ГОСТ | Межгосударственный стандарт |
| ДГУ | Дизель-генераторная установка |
| ДСП | Древесно-стружечная плита |
| ДТП | Дорожно-транспортные происшествия |
| ДЧС | Департамент по чрезвычайным ситуациям |
| ЕСВМ | Единая система видео-мониторинга |
| ИКТ | Информационно-коммуникационные технологии |
| ИС | Информационная система |
| КПСиСУ ГП РК | Комитет по правовой статистике и специальным учетам Генеральной прокуратуры Республики Казахстан |
| МИО | Местные исполнительные органы |
| МСБ | Малый и средний бизнес |
| МФО | Микрофинансовые организации |
| ОКЭД | Общий классификатор видов экономической деятельности |
| ПВХ | Поливинилхлорид |
| ПНЗ | Пост наблюдения загрязнения атмосферного воздуха |
| протокол TCP/IP | Сетевая модель передачи данных, представленных в цифровом виде. Модель описывает способ передачи данных от источника информации к получателю. |
| ПРТ | Программа развития территории |
| РГП | Республиканское государственное предприятие |
| РК | Республика Казахстан |
| СЗПТ | Социально-значимые продовольственные товары |
| СЭР | Социально-экономическое развитие |
| СКУД | Система контроля и управления доступом |
| СНиП РК | Строительные нормы и правила Республики Казахстан |
| ТЗ | Техническое задание |
| технологии IP и PoE | Power over Ethernet (PoE) — технология, позволяющая передавать удалённому устройству электрическую энергию вместе с данными через стандартную витую пару в сети Ethernet. |
| УО | Управление образования города Алматы |
| УК | Управление культуры города Алматы |
| УЗЭ | Управление зеленой экономики города Алматы |
| ЧС | Чрезвычайная ситуация |
| Android | Операционная система мобильных устройств |
| API | API (от англ. application programming interface — «интерфейс программирования приложения») — программный интерфейс, то есть описание способов взаимодействия одной компьютерной программы с другими. |

Алматы қаласы цифрлық даму басқармасы\Әдәмбай А. Ә.

| Термин | Определение |
|---|---|
| Backend | Бэкенд — это разработка бизнес-логики продукта (сайта или веб-приложения), которая отвечает за взаимодействие пользователя с внутренними данными, которые потом отображает фронтенд. |
| BI | BI (от англ. Business Intelligence, «бизнес-интеллект») — это набор средств, методологий и концепций для сбора, обработки и анализа данных, поступающих как из самой компании, так и из внешних источников. |
| COVID-19 | Коронавирусная инфекция |
| ETL | ETL (от англ. Extract, Transform, Load — «извлечение, преобразование, загрузка») — один из основных процессов в управлении хранилищами данных. |
| Frontend | Фронтенд (от англ. Front-end — «передний конец») — это визуальная часть веб-сайта, которую пользователь видит и с которой может взаимодействовать при помощи браузера. |
| iframe | iframe - это тег, который позволяет внутри другого HTML-документа отображать другой. В результате при запуске страницы, на которой присутствует фрейм, фоново подгружается еще одна самостоятельная страница. |
| iOS | Операционная система мобильных устройств |
| MAC адреса | MAC-адрес (от англ. Media Access Control — управление доступом к среде, также Hardware Address, также физический адрес) — уникальный идентификатор, присваиваемый каждой единице активного оборудования или некоторым их интерфейсам в компьютерных сетях Ethernet. |
| SFP модуль | SFP (англ. Small Form-factor Pluggable) — промышленный стандарт модульных компактных приёмопередатчиков (трансиверов), используемых для передачи и приема данных в телекоммуникациях. |

Алматы қаласы цифрландыру басқармасы/Адамбай А. Ә.

# 1. Общие сведения

Отчет подготовлен в рамках договора №21 от 26 мая 2025 года (далее – Договор), заключенного между КГУ «Управление цифровизации города Алматы» и АО «Центр развития города Алматы» на оказание услуги по предоставлению данных системы мониторинга индикаторов развития города Алматы (Ситуационный центр).

Целью услуги является предоставление доступа к Ситуационному центру всем заинтересованным лицам с фокусом на обеспечение потребности:

– акимату, государственным органам города Алматы - для осуществления мониторинга, анализа социально-экономического развития города Алматы на основе различных данных с визуализацией результатов и формированием отчетности в режиме реального времени (онлайн);

– жителям города Алматы – предоставление открытых данных, статистической и аналитической информации, динамического статуса развития города, сервисов и др.

В рамках указанной цели, оказываются следующие виды услуг:

1) Сопровождение модулей Ситуационного центра.

Сопровождение модулей Ситуационного центра обеспечивается в соответствии с пунктом 5.1. Технической спецификации к Договору.

2) Доступ к Ситуационному центру и поддерживающие административные услуги направлений Smart Almaty, предоставление аналитических справок по наиболее ключевым запросам со стороны УЦ на основании данных Ситуационного Центра и других данных, предоставленных УЦ, осуществляется в соответствии с пунктом 5.2. Технической спецификации к Договору.

3) Обеспечение хранения данных, осуществляется в соответствии с пунктом 5.3. Технической спецификации к Договору.

По итогам 4 квартала 2025 года были проведены следующие работы в соответствии с пунктом 5.1. Технической спецификации к Договору:

1) Сопровождение модулей Ситуационного центра.

- Инвентаризация
  - Добавление новых слоев в модуль «Инфраструктура города»:
    - Сетка дорог города;
    - Карта пробок и узких мест;
    - Маршруты общественного транспорта;
    - Пешеходные зоны;
- Мобильное приложение
  - Разработка мобильного приложения (задачи 4 кв.):
    - Предложение инициатив.
    - Умная благотворительность/фудшеринг.

Алматы қаласы цифрландыру басқармасы Адаңбай А. Ә.

- Мониторинг районов.

– Разработка информационный и ГИС модулей
  - Разработка модуля «Мониторинг сейсмоустойчивости городских объектов» (задачи 4 кв.)
  - Разработка модуля «Мониторинг сейсмической опасности» (задачи 3 кв.)
  - Разработка модуля «Карта туристических объектов города Алматы» (задачи 3 кв.)
  - Разработка Сервиса «Цифровой рейтинг школ» (задачи 3 кв.)
  - Разработка модуля «Прогноз дефицита мест в школах» (задачи 3 кв.)
  - Разработка модуля «Территориальное распределение поликлиник»
  - Разработка модуля «Карта нозологий и смертности»
  - Разработка модуля «Мониторинг госпитализаций и коечной мощности»
  - Разработка модуля «Отображение проектов на карте»
– Разработка аналитических модулей
  - Расположение мест сбора при ЧС
  - Развитие Аналитики преступности
  - Анализ парков и скверов (задачи 3 кв.)
  - Оценка транспортной доступности общественных пространств (задачи 3 кв.)
  - Развитие инфраструктуры для отдыха и спорта (задачи 3 кв.)
  - Развитие модуля «Аналитика нагрузки дорог» (задачи 3 кв.)
– Дата-инженерия
  - Сопровождение ЕХД
  - Сопровождение модуля ЕСВМ
  - Интеграция с Damubala и Bilimalmaty (задачи 3 кв.)

2) Доступ к Ситуационному центру и поддерживающие административные услуги Smart Almaty.

В рамках 4 квартала был предоставлен доступ к Ситуационному центру, выполнены работы по ведению обновления и актуализации данных, а также предоставлены услуги по администрированию проектов Smart Almaty.

3) Обеспечение хранения данных.

В рамках 4 квартала было обеспечено размещение оборудования в общем количестве 189 штук, занявших 432 U* (юнитах).

Также был предоставлен доступ к двум облачным хранилищам на базе платформы Nextcloud с сопровождением и администрированием.

Алматы қаласы инфрақұрылым басқармасы\АДенгібай А. Ә.

# 2. Сопровождение модулей Ситуационного центра

Работы проводились в рамках пункта 5.1 Технической спецификации к Договору.

## 2.1. Инвентаризация. Модуль «Инфраструктура города»

Модуль обеспечивает визуализацию данных в виде слоев ГИС по категориям объектов. Объекты включают в себя паспортные данные.

**Общие задачи разработки на ГИС-карте на 2025 год:**

1) Сбор данных: ручной, взаимодействие с гос.органами, парсеры открытых источников.
2) Вычистка данных и загрузка в БД.
3) Обновление функционального блока категорий.
4) Дополнение недостающих полей и данных по слоям.
5) Оптимизация дизайна слоев.
6) Разработка системы со стороны backend.
7) Доработка структуры данных.
8) Доработка со стороны frontend.
9) Приведение к стандарту карточек и логики отображения.
10) Тех.работы по оптимизации быстродействия и нагрузочных мощностей системы.

**Результаты работ за 4 квартал:**

Добавление слоев для аналитики:

- Цены на недвижимость;
- Промышленные объекты;
- Территориальное распределение по школам;
- Территориальное распределение по поликлиникам.

### 2.1.1. Цены на недвижимость

В рамках выполнения задач было выполнена разработка комплексных парсеров для сбора данных с сайта Krisha.kz. Целью работы стало автоматизированное получение актуальной информации по различным категориям недвижимости для последующего анализа.

Всего было создано 12 парсеров, охватывающих категории аренды и продажи, в нынешнем квартале было создано 7 парсеров:

Аренда:

- гаражи и паркинги;
- промбазы и заводы;
- аренда комнат.

Продажа:

- бизнес;
- гаражи и паркинги;

Алматы қаласы цифрландыру басқармасы\АДәндібой А. Ә.

- промбазы и заводы;
- участки.

Каждый парсер включает два основных модуля: быстрый сбор объявлений со страниц каталога и детальный сбор данных с карточек объектов.

Особенности реализации

Быстрый парсер для аренды комнат выполняет определение количества страниц и последующий обход всех объявлений. Он собирает основную информацию:

- ID и UUID;
- ссылку на объявление;
- цену;
- адрес;
- краткое описание;
- количество изображений и путь к каталогу изображений;
- дату и время парсинга.

Модуль использует случайный User-Agent и задержки между запросами, что снижает риск блокировок. Страницы обходятся в обратном порядке для получения более свежих данных. Данные сохраняются после каждого объекта, что исключает потерю информации при сбое. Ведется файл с обработанными страницами, позволяющий пропускать уже собранные данные.

Детальный парсер выполняет углубленный сбор информации по каждому объявлению и извлекает расширенный набор характеристик:

- параметры объекта (этаж, площадь, тип здания, ремонт, год постройки);
- условия проживания (мебель, интернет, охрана, телефон, санузел, балкон, парковка);
- параметры из блока offer__parameters: особенности безопасности, время заселения, условия проживания, наличие залога;
- полное текстовое описание объявления;
- скрытые данные из скриптов страницы.

Каждое объявление сохраняется незамедлительно после обработки, ведется файл собранных ссылок, что обеспечивает устойчивость к перезапускам. Также используются задержки между запросами и вывод индикатора прогресса.

Алматы қаласы цифрландыру басқармасы\Дәнібай А. Ә.

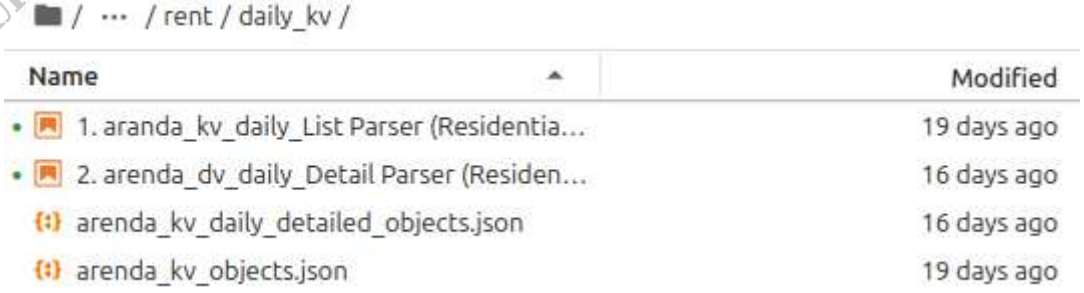
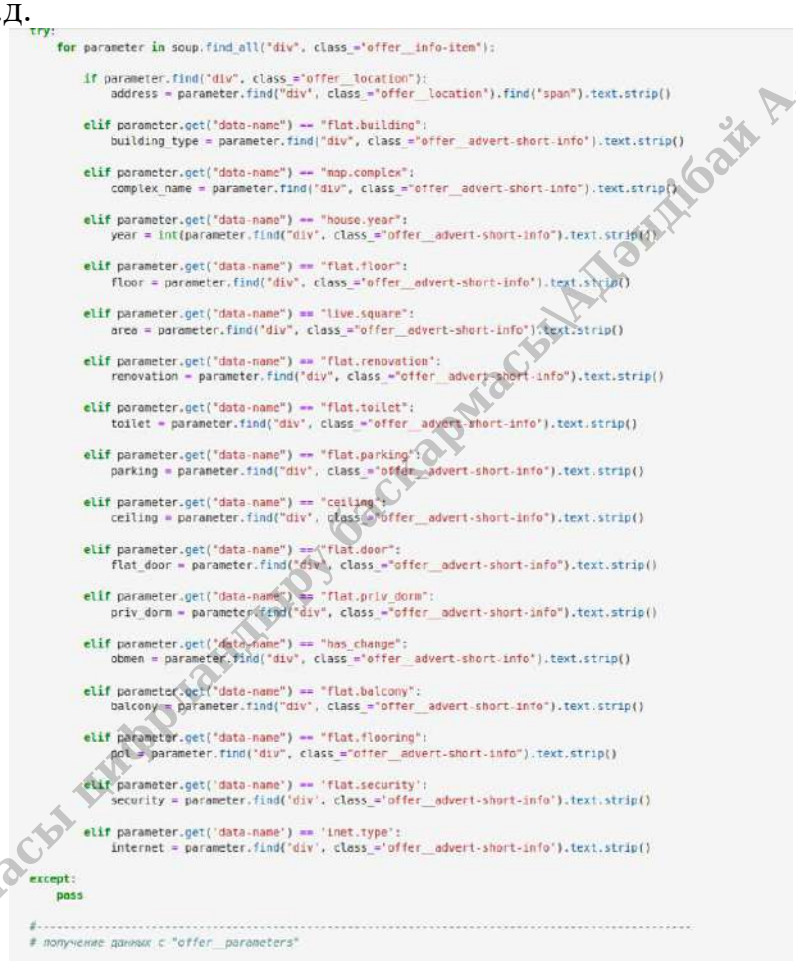



Рисунок – 2.1.1.1. Парсеры по аренде комнат и сохраненные данные

В ходе работы над задачей была создана масштабируемая архитектура парсеров, охватывающая 12 категорий недвижимости. Реализованы механизмы защиты от дублирования, сохранение данных в реальном времени, устойчивость к сбоям, работа через прокси и структурированный сбор данных как на уровне списка объявлений, так и на уровне детальных карточек.

Парсеры готовы к расширению и адаптации под новые категории и аналитические задачи. Находятся они на сервере и в сетевой папке.

Так же некоторые парсеры были заново созданные которые были выполнены в прошлых кварталах, так как сайт krisha.kz изменил тело структуры сайта, домены и т.д.



```
try:
    for parameter in soup.find_all("div", class_="offer__info-item"):

        if parameter.find("div", class_="offer__location"):
            address = parameter.find("div", class_="offer__location").find("span").text.strip()

        elif parameter.get("data-name") == "flat.building":
            building_type = parameter.find("div", class_="offer__advert-short-info").text.strip()

        elif parameter.get("data-name") == "map.complex":
            complex_name = parameter.find("div", class_="offer__advert-short-info").text.strip()

        elif parameter.get("data-name") == "house.year":
            year = int(parameter.find("div", class_="offer__advert-short-info").text.strip())

        elif parameter.get("data-name") == "flat.floor":
            floor = parameter.find("div", class_="offer__advert-short-info").text.strip()

        elif parameter.get("data-name") == "live.square":
            area = parameter.find("div", class_="offer__advert-short-info").text.strip()

        elif parameter.get("data-name") == "flat.renovation":
            renovation = parameter.find("div", class_="offer__advert-short-info").text.strip()

        elif parameter.get("data-name") == "flat.toilet":
            toilet = parameter.find("div", class_="offer__advert-short-info").text.strip()

        elif parameter.get("data-name") == "flat.parking":
            parking = parameter.find("div", class_="offer__advert-short-info").text.strip()

        elif parameter.get("data-name") == "ceiling":
            ceiling = parameter.find("div", class_="offer__advert-short-info").text.strip()

        elif parameter.get("data-name") == "flat.door":
            flat_door = parameter.find("div", class_="offer__advert-short-info").text.strip()

        elif parameter.get("data-name") == "flat.priv_dorm":
            priv_dorm = parameter.find("div", class_="offer__advert-short-info").text.strip()

        elif parameter.get("data-name") == "has_change":
            obmen = parameter.find("div", class_="offer__advert-short-info").text.strip()

        elif parameter.get("data-name") == "flat.balcony":
            balcony = parameter.find("div", class_="offer__advert-short-info").text.strip()

        elif parameter.get("data-name") == "flat.flooring":
            pol = parameter.find("div", class_="offer__advert-short-info").text.strip()

        elif parameter.get('data-name') == 'flat.security':
            security = parameter.find('div', class_='offer__advert-short-info').text.strip()

        elif parameter.get('data-name') == 'inet.type':
            internet = parameter.find('div', class_='offer__advert-short-info').text.strip()

except:
    pass

#-------------------------------------------------------------------------------------------
# получение данных с "offer__parameters"
```

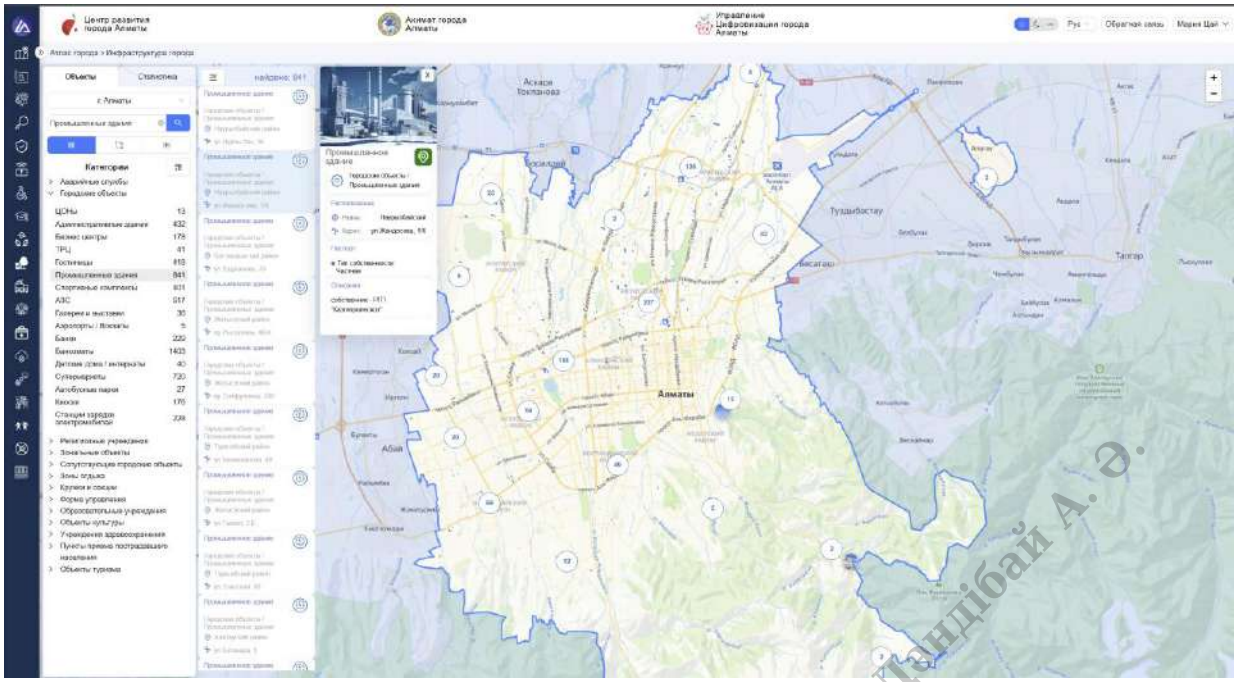
Рисунок – 2.1.1.2. Детальный парсер по аренде комнат

### 2.1.2. Промышленные объекты

В рамках выполнения задач было выполнена разработка комплексных парсеров для сбора данных с сайта 2ГИС. Целью работы стало автоматизированное получение актуальной информации по различным категориям промышленных объектов для последующего анализа.

Результатом работы стал слой «Промышленные здания» с 841 объектом.

Алматы қаласы цифрландыру басқармасы\Алаідбай А. Ә.



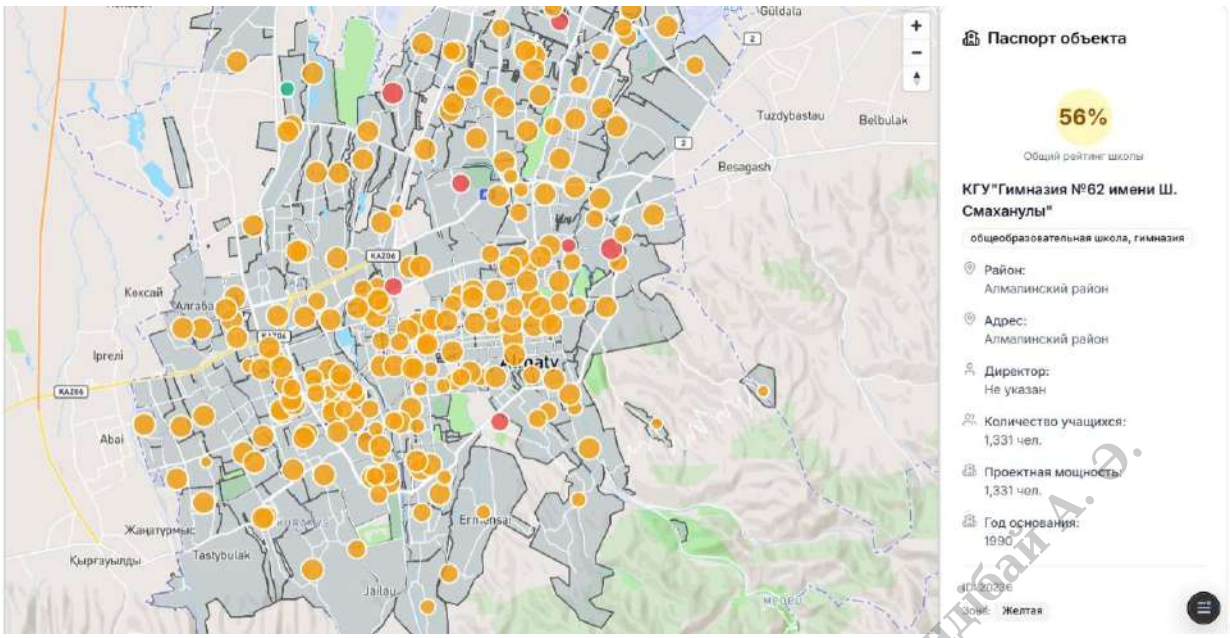
### 2.1.3. Территориальное распределение по школам

В рамках формирования геоаналитической базы данных Ситуационного центра проведены работы по уточнению и визуализации территориального распределения жилых домов по закреплённым школам города. Управлением образования были предоставлены исходные списки домов, закреплённых за каждой школой. Полученные данные прошли предварительную верификацию и сопоставление с актуальными адресными сведениями.

На следующем этапе проведено ручное нанесение соответствующих объектов на картографическую основу с использованием ГИС-инструментов. Для каждого дома определена принадлежность к конкретной школе, после чего сформирован специализированный геослой, содержащий пространственные привязки, атрибутивную информацию и уникальные идентификаторы объектов.

Сформированный геослой интегрирован в модуль «Инфраструктура» и включён в единое хранилище данных для последующего применения в аналитических процессах. На его основе выполняются расчёты шаговой доступности, моделирование нагрузок на школы, прогнозирование потребности в строительстве новых образовательных объектов и другие задачи дата-аналитики.

Алматы қаласы цифрландыру басқармасы АДандібай А. Ә.



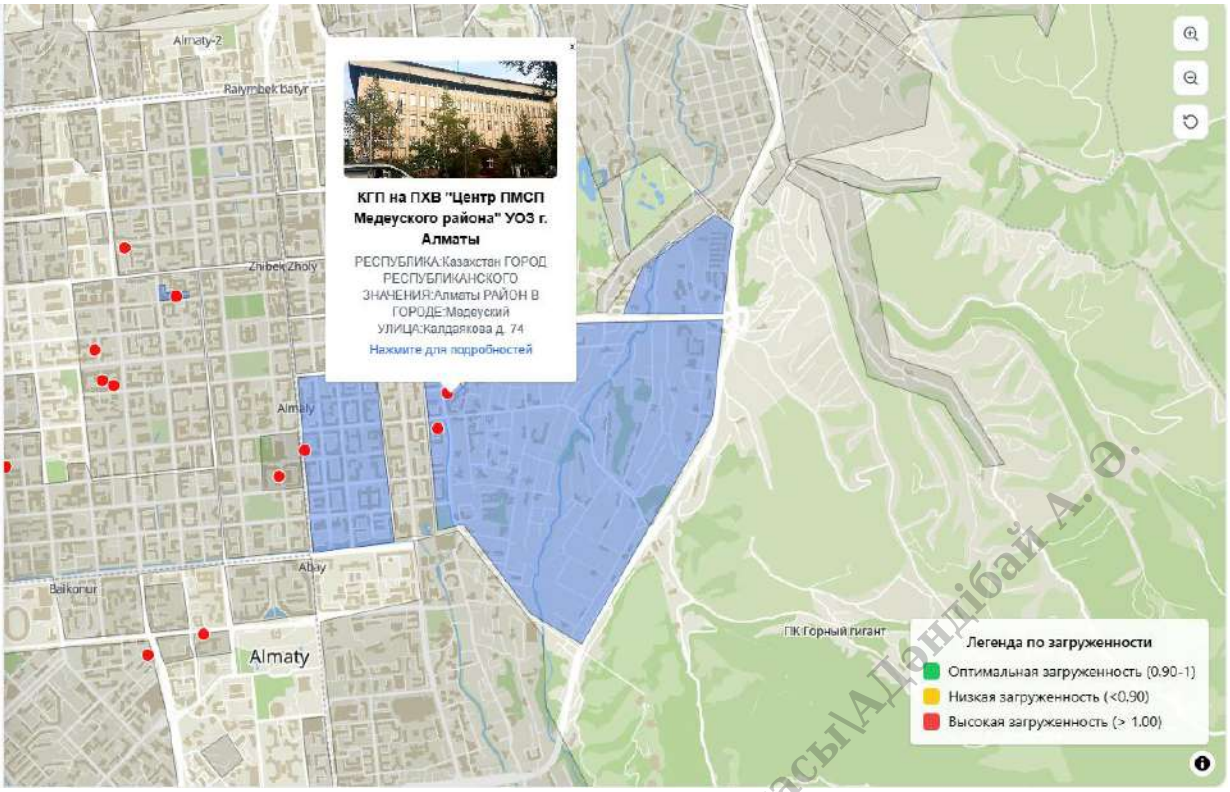

### 2.1.4. Территориальное распределение по поликлиникам

Для формирования единой геоаналитической базы Ситуационного центра проведены работы по уточнению территориального прикрепления жилых домов к поликлиникам города. Управлением общественного здравоохранения были предоставлены официальные списки домов, закреплённых за каждой поликлиникой. Полученные сведения прошли первичную проверку, нормализацию адресов и сопоставление с существующей адресной структурой города.

После верификации данных выполнено ручное нанесение всех объектов на картографическую основу с применением ГИС-инструментов. Для каждого жилого дома определена его принадлежность к конкретной поликлинике, после чего сформирован специализированный геослой, включающий координатные привязки, атрибутивные данные и уникальные идентификаторы.

Сформированный геослой интегрирован в модуль «Инфраструктура» и добавлен в единое хранилище пространственных данных. Данный слой применяется в аналитических расчётах по шаговой доступности медицинских услуг, моделированию нагрузки на поликлиники, оценке равномерности охвата населения медицинской инфраструктурой, а также в прогнозных моделях развития сети объектов здравоохранения.

Алматы қаласы цифрландыру басқармасы Адәндібай А. Ә.




Алматы қаласы цифрландыру басқармасы\Адильбай А. Ә.

## 2.2.Разработка мобильного приложения

**Цель**:

Разработать мобильное приложение, которое предоставит гражданам доступ к текущим сервисам Ситуационного центра города Алматы и новому сервису, направленному на цифровую инклюзию для лиц с инвалидностью (ЛСИ).

**Этапы разработки:**

4-й квартал:

- Разработка сервиса цифровой инклюзии
- Разработка административной панели для обратной связи от пользователей
- Тестирование пользовательских сценариев.
- Публикация приложения в Google Play и App Store.

**Заказчик:** Проектный офис ГРБН «Инклюзивная среда, образование и медицина», Управление занятости и социальных программ

### 2.2.1. Разработка сервиса цифровой инклюзии

В рамках выполнения задач 4-квартала по модулю «Цифровая инклюзия» на стороне backend были проведены работы по созданию основы, которая включает следующие пункты:

- Функция авторизации – разработан механизм авторизации для пользователей мобильного приложения;
- API для главной страницы и карты - созданы эндпоинты для вывода

агрегированных данных на главный экран и для отображения гео-объектов на карте;

– API для фильтрации - реализован функционал фильтрации данных по различным параметрам, что позволяет пользователям мобильного приложения получать кастомизированные выборки данных.

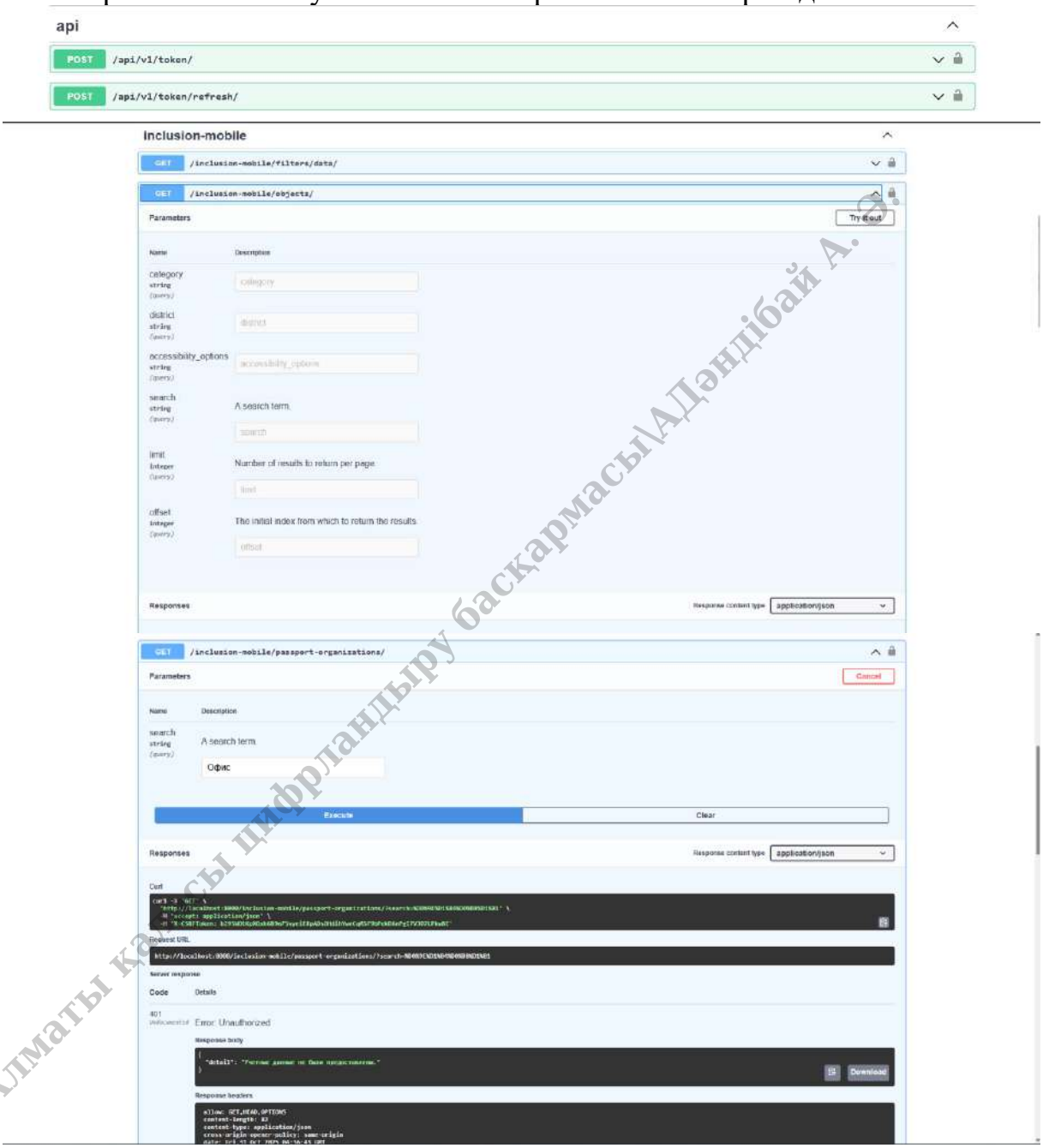



Рисунок – 2.2.1.1. Пример API для модуля «Цифровая инклюзия»

В части функции авторизации был создан безопасный модуль входа, основанный на корректной обработке access и refresh токенов. Это позволило обеспечить стабильную авторизацию и мгновенное перенаправление пользователя на главный экран.

Были проведены работы по интеграции Яндекс.Карты и реализован

Алматы қаласы цифрландыру басқармасы\АДәндібай А. Ә.

механизм получения координат объектов с бэкенда: приложение автоматически загружает массив объектов, размещает их на карте и позволяет пользователю взаимодействовать с точками через удобный pop-up элемент с данными об объекте, его категории, адресе, телефоне и уровне доступности.

Большое внимание было уделено реализации системы фильтров, которая позволяет выбирать объекты по следующим параметрам:

- Категория;
- Район;
- Признак доступности;
- Наличие элементов МГН;
- Дополнительные параметры.

Каждый фильтр раскрывается во вложенные уровни, предоставляя пользователю возможность точно настроить параметры поиска.

Несмотря на многоуровневую структуру, интерфейс остаётся лёгким, понятным и не перегруженным. Такой подход делает навигацию по объектам гибкой и эффективной

Также в сервис был внедрен отдельный раздел «Профиль», который является стал личным рабочим пространством пользователя. В разделе отображаются поданные заявки, черновики и существующие паспорта объектов.

Один из самых объёмных функций, которое было внедрено в сервис это - создание и редактирование нового паспорта объекта.

Процесс создание нового паспорта включает ввод следующей информации:

- Юридическая информация об объекте;
- данные о здании;
- координат места на карте;
- информацию по виду деятельности;
- и других параметров.

Также был разработан огромный комплекс форм по оценке доступности:

- входных групп;
- путей движения;
- санитарных помещений;
- ТВК;
- зон обслуживания;
- и многое другое.

Каждый раздел содержит подпункты, фактические значения, рекомендации и фотофиксацию. Благодаря такому уровню детализации система становится удобным инструментом для профессиональной проверки объектов.

Важной частью интерфейса стали избранные объекты. Была реализована логика добавления и удаления объекта из избранного списка через эндпоинты API, обеспечив мгновенное визуальное переключение состояния иконки сердечка. Это создаёт ощущение живости интерфейса и делает работу

Алматы қаласы цифрландыру басқармасы\АДаирбай А. Ә.

пользователя более комфортной.

В итоге приложение объединило в себе современную архитектуру, глубокую техническую проработку и плавное пользовательское взаимодействие.

Каждый модуль — авторизация, карты, фильтры, профиль, паспорта, доступность, избранное — был создан с акцентом на надёжность, удобство и функциональную ценность.

Сервис «Цифровая инклюзия» стал мощным инструментом для работы со сведениями об объектах и оценкой доступности городской среды.

***Разработка и внедрение экрана авторизации***

В рамках подготовки мобильного приложения «Цифровая инклюзия» был реализован и оформлен полноценный экран авторизации (Login), являющийся точкой входа пользователя в систему.

Основной задачей этапа стало создание стабильного, интуитивного и визуально выдержанного интерфейса, который соответствует общему стилю проекта и обеспечивает безопасный доступ к функционалу приложения.

Для повышения удобства и UX была разработана адаптивная верстка страницы, корректно отображающаяся на устройствах разных размеров. Дополнительно были применены единые дизайн-компоненты (цветовые палитры, стили кнопок, текстовые поля), обеспечивающие консистентность интерфейса на всём протяжении приложения.

Особое внимание было уделено логике работы авторизации. На данном этапе была реализована базовая форма ввода с валидацией:

- Поле имени пользователя (логин) — принимает электронную почту, добавлены проверки формата и состояния ввода.
- Поле пароля — реализована защита ввода, индикатор скрытия/показа символов.
- Кнопка «Вход» — активируется только при валидных данных, что снижает количество ошибочных попыток входа.

Страница дополнена текстовым блоком с ключевым позиционированием проекта: «Цифровая инклюзия — город, в котором каждому удобно. Навигация, сервисы и доступность в одном приложении». Это усиливает восприятие платформы и задаёт корректный контекст перед входом.

Алматы қаласы цифрландыру басқармасы Цандібай А. С.

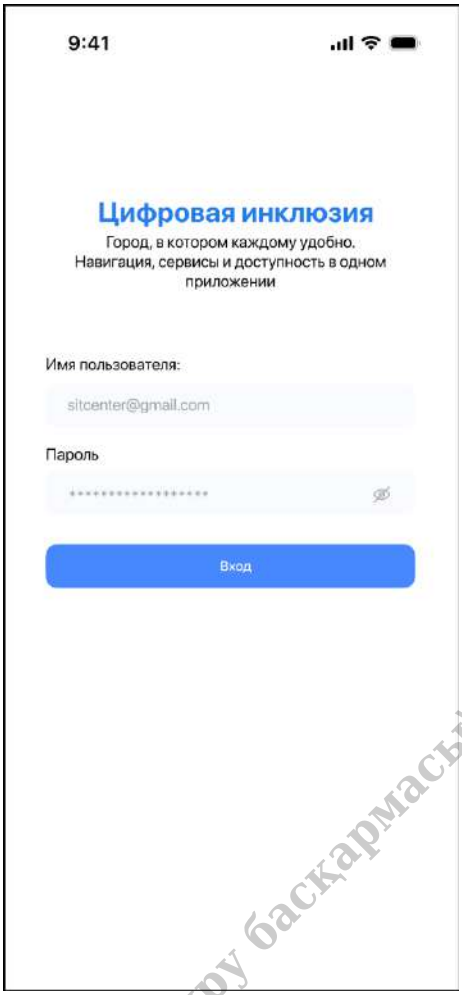



Рисунок – 2.2.1.2. Функция авторизации в системе

***Разработка и внедрение многоуровневой функции фильтрации данных***

В ходе реализации функционала навигации и поиска доступных объектов в рамках мобильного приложения «Цифровая инклюзия» была разработана и внедрена полнофункциональная система фильтрации объектов городской инфраструктуры.

Данный модуль стал ключевым элементом пользовательского сценария, обеспечивающим точное и удобное формирование выборки объектов по нескольким параметрам.

Основной целью этапа являлось создание гибкого, интерактивного и интуитивно понятного инструмента, который позволяет пользователю легко отбирать объекты по критериям категории, района, доступности и набора элементов доступной среды.

Экран фильтра представляет собой многоуровневую структуру, включающую несколько выпадающих списков (Dropdown), каждый из которых раскрывает соответствующие параметры:

1. Категория объекта:

Ақпараттық цифрландыру басқармасы\АДәндібай А. Ә.

• административные здания;
• ЦОНы;
• бизнес-центры;
• торговые центры;
• образовательные учреждения;
• учреждения здравоохранения;
• промышленные здания.

2. Район:
• Алатауский;
• Ауэзовский;
• Алмалинский;
• Бостандыкский;
• Жетысуский;
• Медеуский;
• Наурызбайский;
• Турксибский.

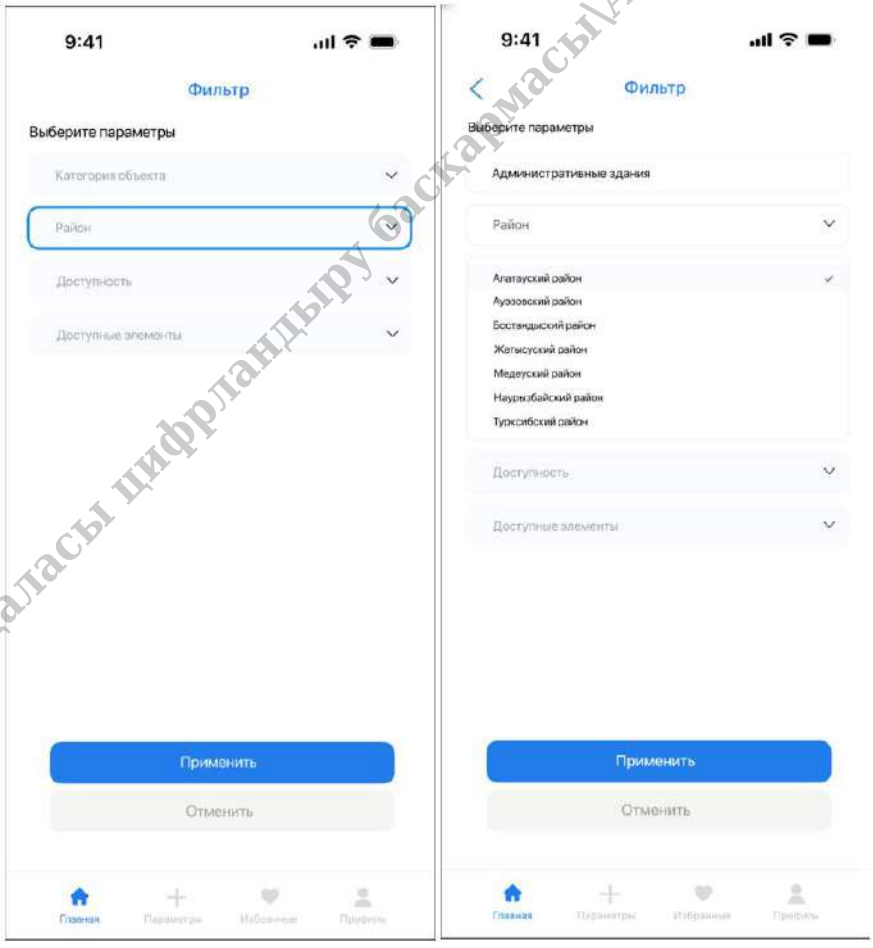



Рисунок – 2.2.1.3. Функция фильтрации данных. Районы.

3. Доступность (типы доступных элементов инфраструктуры)

Этот блок включает более сложную структуру данных и используется для отображения набора параметров доступности:

Алматы қаласы цифрландыру басқармасы\АДәндібай А. Ә.

- входная группа;
- пути движения;
- зона обслуживания;
- санитарно-бытовые помещения;
- средства информации и телекоммуникации;
- территория объекта.

Параметры отображаются в виде выпадающих списков, поддерживают множественный выбор и синхронизацию состояния, что позволяет комбинировать сразу несколько признаков доступности.

4. Доступные элементы

Данный уровень представляет собой наиболее детализированные и социально ориентированные параметры.

Здесь пользователю предлагается выбрать элементы доступности для разных групп населения:

- ♿ “ K” — люди, передвигающиеся на кресле-коляске;
- 🚶 “O” — люди, с поражением опорно-двигательного аппарата;
- 👁 “Z” — люди с нарушением зрения;
- 🦻 “C” — люди с нарушением слуха.

Каждый пункт снабжён чекбоксом, который поддерживает многократный выбор, имеет отдельное состояние, а также визуальную индикацию выбранных параметров.

Данный блок требует правильной структуры данных, поэтому при реализации была применена оптимизированная модель, позволяющая корректно отображать большое количество элементов без потери производительности.

***Разработка и интеграция раздела «Мой профиль»***

Функциональный раздел «Мой профиль» является ключевым компонентом пользовательской части приложения «Цифровая инклюзия», обеспечивающим персонализацию и доступ к данным, связанным с объектами и регистрациями пользователя.

На данном этапе была выполнена комплексная реализация экранов профиля, включая управление заявками, просмотр черновиков и переход к паспорту объекта.

Цель разработки — создать простой и интуитивный интерфейс, который позволит пользователю оперативно получать доступ к персональной информации, следить за статусами заявок и управлять данными объектов, размещённых им в системе.

Раздел состоит из трёх основных частей:

**1. Паспорт объекта**

Переход в данный раздел открывает функциональность, связанную с управлением объектами, закреплёнными за пользователем.

В текущей итерации логика связана с отображением базовых параметров

Алматы қаласы цифрландыру басқармасы\АДәнділбай А. О.

объекта и формы его паспортизации.

Подготовлен UI-каркас и навигационная структура для дальнейшей интеграции с backend-модулем паспорта объекта.

**2. Заявки на регистрацию**

Это основной функциональный блок, над которым было проделано наиболее объёмное количество работы.

После выбора пункта «Заявка на регистрацию» пользователю отображается список всех поданных им заявок. Каждая заявка отображается в виде карточки (Card), включающей:

- наименование объекта;
- дату подачи заявки;
- статус обработки (например: «Паспорт отправлен», «На рассмотрении», «Одобрено»).

Карточки имеют визуально выделенный стиль, лёгкое скругление, тени и отступы, что обеспечивает хорошую читаемость и соответствует общей стилистике приложения.

**3. Черновики**

Раздел предназначен для работы с сохранёнными, но ещё не отправленными заявками.

Пользователь может вернуться к редактированию и завершить оформление регистрации объекта позже.

На данном этапе была подготовлена UI-структура, связанная с отображением черновиков, и архитектурный каркас для будущей реализации CRUD-операций.

**4. Навигация и взаимодействие**

При нажатии на карточку заявки в разделе «Заявки на регистрацию» происходит переход на детальный экран, который содержит:

- название объекта;
- дату регистрации;
- статус;
- дополнительные данные (по мере расширения API).

Это обеспечивает пользователю полный контроль над текущими заявками и полную прозрачность процесса.

Для удобства навигации предусмотрена кнопка возврата в верхней панели, а также корректная интеграция в нижнее меню навигации приложения (Bottom Navigation Bar).

Алматы қаласы цифрландыру басқармасы\АДәлібай А. Ә.

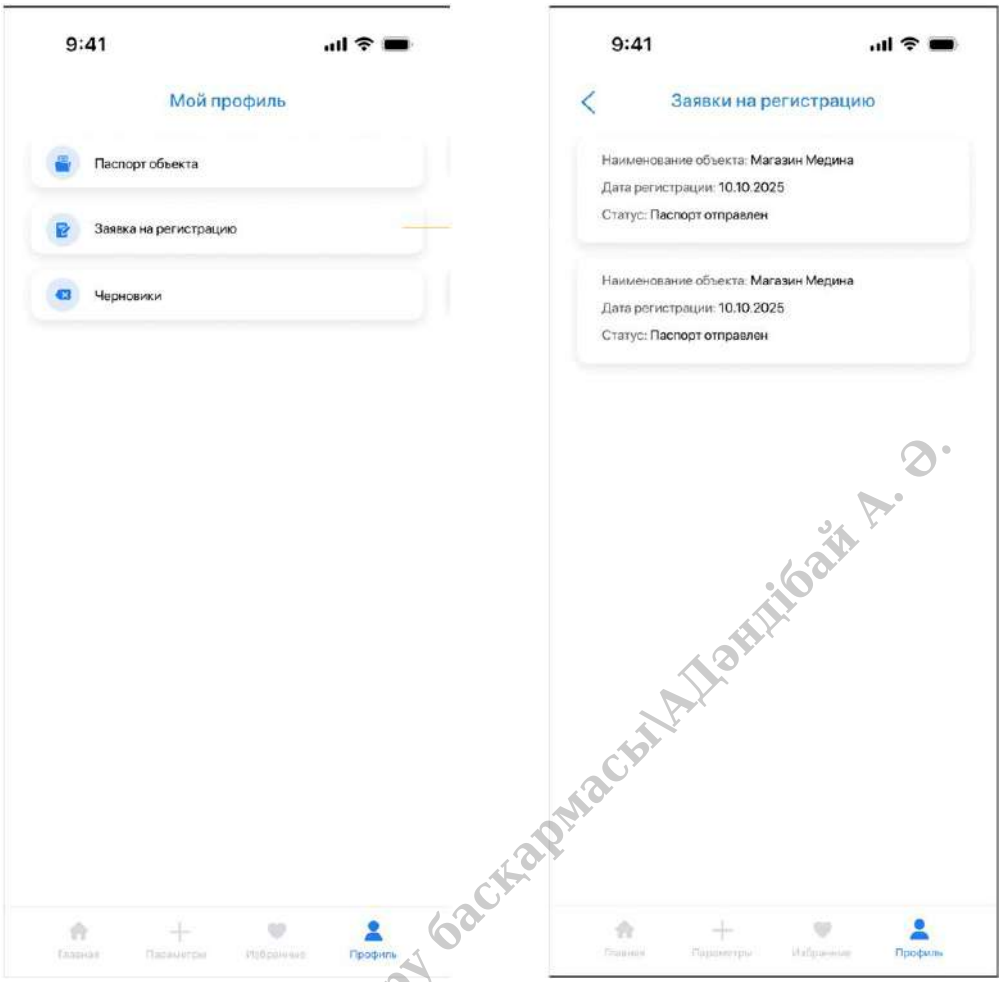



Рисунок – 2.2.1.4. Мой профиль

Алматы қаласы цифрландыру басқармасы\АДәндібай А. Ә.

***UI/UX-решения***

При разработке интерфейсов профиля использовались следующие подходы:

- **Минималистичный дизайн**, основанный на чистых поверхностях и простых формах.
- **Контрастная типографика** для повышения читаемости статусов и названий объектов.
- **Единые компоненты карточек**, позволяющие масштабировать функционал без усложнения UI.
- **Плавная навигация** без скрытых действий, что делает модуль прозрачным и удобным даже для пользователей с минимальным опытом работы с мобильными приложениями.

***Реализация раздела «Избранные»***

Функциональный модуль «Избранные» служит важной частью пользовательской экосистемы сервиса «Цифровая инклюзия», обеспечивая быстрый доступ к объектам, которые пользователь отметил, как наиболее интересные, полезные или требующие регулярного просмотра.

Данный раздел приложения повышает удобство навигации, ускоряет повторный доступ и формирует персонализированный опыт работы с городской инфраструктурой.

Основная цель разработки данного модуля заключалась в создании удобного интерфейса для отображения избранных объектов, обеспечении корректной синхронизации данных с сервером и реализации удобного взаимодействия с карточкой объекта.

Экран «Избранные» представляет собой список карточек, каждая из которых содержит ключевую информацию о сохранённом объекте.

При разработке были реализованы следующие элементы:

1. Карточка объекта

Каждая карточка включает:

- Наименование объекта (например: «Городская больница №7»);
- Категорию объекта (учреждение здравоохранения, образовательное учреждение, административное здание и др.);
- Адрес в формате «район – улица – номер дома»;
- Контактный телефон, если он доступен в данных объекта;
- Доступные элементы — перечень параметров доступной среды (например: территория объекта, входная группа);
- Общий уровень доступности для людей с ограниченными возможностями (например: «частично доступен», «полностью доступен»);
- Иконку избранного (сердце) — синюю при активном состоянии.

Карточка имеет визуально выделенный минималистичный стиль с мягкими тенями, округлёнными углами и чёткой структурой текста. Такой подход обеспечивает отличную читаемость и эстетическую гармонию с другими экранами приложения.

2. Удаление объекта из избранного

Иконка "сердца" является интерактивной:

- при повторном нажатии объект удаляется из избранного;
- изменения моментально отражаются в UI;
- состояние синхронизируется с локальным хранилищем и (при наличии API) с сервером.

Это создаёт удобный и быстрый механизм управления списком избранных объектов.

***UI/UX-решения***

При проектировании раздела были использованы принципы доступного и лёгкого взаимодействия:

- максимальная простота интерфейса — только содержательная информация, без лишних элементов;
- удобство просмотра — структурирование данных по строкам, визуальные разделители, ровное выравнивание текста;

Алматы қаласы цифрландыру басқармасы Адамбай А. Ә.

• фокус на ключевых данных — адрес, доступность, доступные элементы и контакты;

• единообразие дизайна — карточка выполнена в стиле других модулей («Профиль», «Фильтр»), что обеспечивает целостность интерфейса.

Рисунок – 2.2.1.5. Избранные

***Разработка комплексного модуля формирования паспорта объекта***

В рамках проекта был реализован многоэтапный модуль формирования паспорта объекта, предназначенный для структурированного сбора данных, оценки доступности инфраструктуры и подготовки стандартизированной информации для государственных сервисов.

Модуль покрывает полный цикл ввода данных — от выбора типа паспорта до формирования детализированного отчёта с фотофиксацией, рекомендациями и метрическими значениями.

Модуль решает сразу несколько задач:

- обеспечивает единый унифицированный интерфейс для создания паспорта объекта;
- стандартизирует ввод сложных параметров, требующих десятки форм и вложенных разделов;
- автоматизирует контроль данных, снижая вероятность ошибок инспекторов и операторов;

Алматы қаласы цифрландыру басқармасы\АДәндібай А. Ә.

• позволяет собирать фотофиксацию по каждому элементу доступности;

• формирует корректную структуру данных для передачи в бэкенд и дальнейшей интеграции с государственными сервисами.

***Основные стадии и реализованные экраны***

1. Выбор типа паспорта

Разработан экран, позволяющий выбрать вид паспорта:

- Паспорт объекта;
- Паспорт гостиницы;
- Паспорт общепита;
- Паспорт доступности мест;
- Паспорт остановок ОТ.

Каждый вариант ведёт к разным наборам форм, что обеспечено динамической конфигурацией интерфейса.

2. Создание нового паспорта

Пользователь переходит к созданию нового объекта:

- отображение статуса;
- фиксация причины отклонений;
- загрузка фотоотчётов;
- ввод юридической информации.

Реализована гибкая система обязательных/необязательных полей, которая подстраивается под тип объекта.

3. Основные характеристики объекта

Большой блок полей:

- адрес, улица, дом;
- количество этажей;
- год постройки;
- район;
- телефоны, email.

Добавлена валидация формата телефонов и e-mail.

4. Определение точки координат

Интерактивная карта:

- выбор точки на карте;
- автоматическое определение адреса;
- фиксация GPS-координат;
- возможность корректировки вручную.

Используется кастомный компонент карты с маркерами и обработкой жестов.

5. Характеристики деятельности

Собраны структурированные данные:

- вид деятельности;
- подвид;
- описание;

Алматы қаласы цифрландыру басқармасы\Аблайхан А. Ә.

- форма собственности;
- объём предоставляемых услуг;
- услуги для людей с ограниченными возможностями.

Блоки адаптируются в зависимости от выбранного типа паспорта.

6. Многообъектная структура территории

Пользователь может добавить один или несколько объектов на территории:

- тип здания;
- количество этажей;
- дополнительные элементы.

Создан удобный интерфейс для добавления множества вложенных сущностей.

7. Секции доступности для МГН

7.1. Территория объекта

- пандусы;
- тактильные зоны;
- указатели;
- бордюры;
- зоны отдыха;
- минимальные проходы;
- фотофиксация каждого элемента;
- метрические значения;
- выбор рекомендации.

Каждый пункт содержит:

- чекбокс «не требуется»;
- фактические значения;
- рекомендации;
- нагрузки по фотофиксации (до 6 фото на элемент).

7.2. Входная группа

Покрыты десятки параметров:

- навесы;
- двери;
- ширина проёмов;
- кнопки вызова;
- пандусы при входе;
- поручни;
- визуальные отметки;
- дублированные тактильные линии;
- фотографии зоны входа.

Формы оптимизированы с учётом большого количества однотипных блоков.

7.3. Пути движения

Проработаны особенности:

Алматы қаласы цифрландыру басқармасы\АДәндібай А. Ә.

- коридоры;
- холлы;
- зоны ожидания;
- лифты;
- ширина проходов;
- уклоны;
- поручни;
- поворотные зоны;
- элементы маршрута;
- фотофиксация.

Каждый пункт снабжён динамическими полями рекомендаций и фактических значений.

7.4. Зона оказания услуг

Учтены детали:

- обслуживание за столом;
- высота стойки;
- обслуживание через окошко;
- доступность для колясочников;
- зоны для сидения;
- кнопки вызова;
- фотофиксация.

Сложная структура адаптируется под разные типы объектов (например, рестораны, госучреждения, гостиницы).

8. Санитарно-бытовые помещения

Реализованы формы проверки:

- доступность санузлов;
- ширина дверных проёмов;
- поручни;
- тактильные полосы;
- специальные зоны для инвалидных колясок;
- высота унитаза;
- возможность бокового подъезда;
- фотофиксация каждой единицы.

Добавлена многоступенчатая загрузка фото.

9. Средства информации и телекоммуникации

Огромная форма для проверки:

- визуальные указатели;
- акустические средства;
- тактильные средства;
- информационные таблички;
- дублирование информации;
- фотофиксация элементов.

Система автоматически скрывает неактуальные блоки, если они не

Алматы қаласы цифрландыру басқармасы\АДәндібай А. О.

требуются.

10. Выбор ответственного органа
    - выбор органа из списка;
    - привязка к объекту;
    - подготовка данных к сохранению.

11. Сохранение

На последнем шаге реализованы две стратегии:

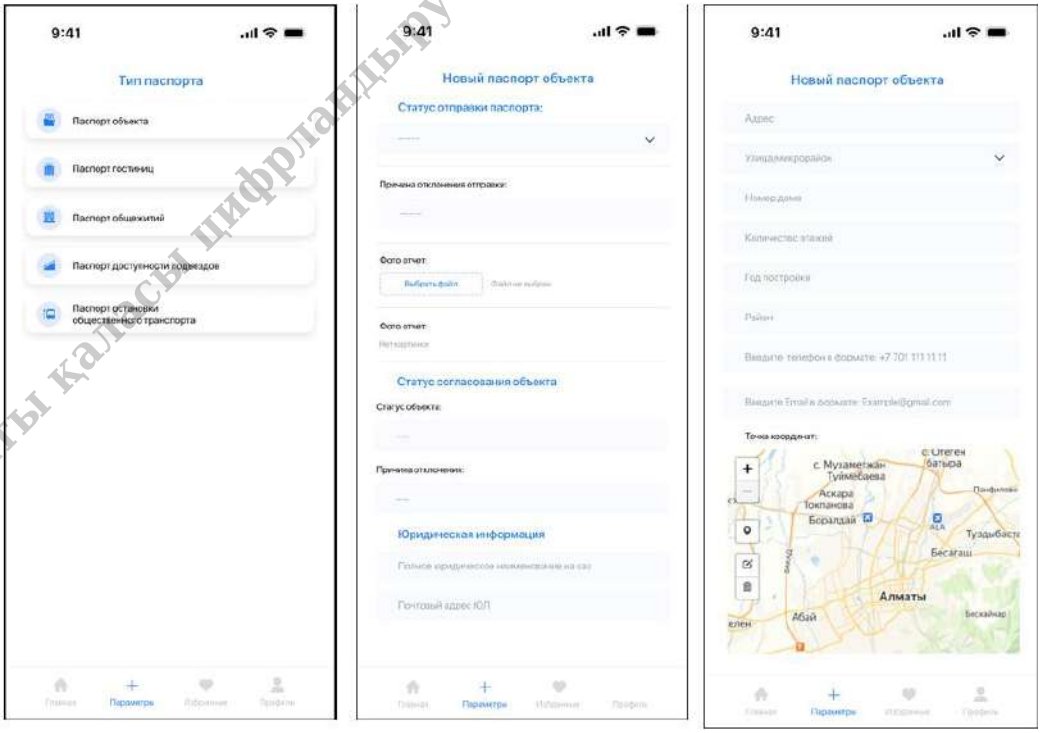
- Сохранить и продолжить;
- Сохранить и добавить новый объект.

Реализовано корректное формирование тел запросов для бэкенда, включая вложенные структуры, массивы фото и рекомендации.

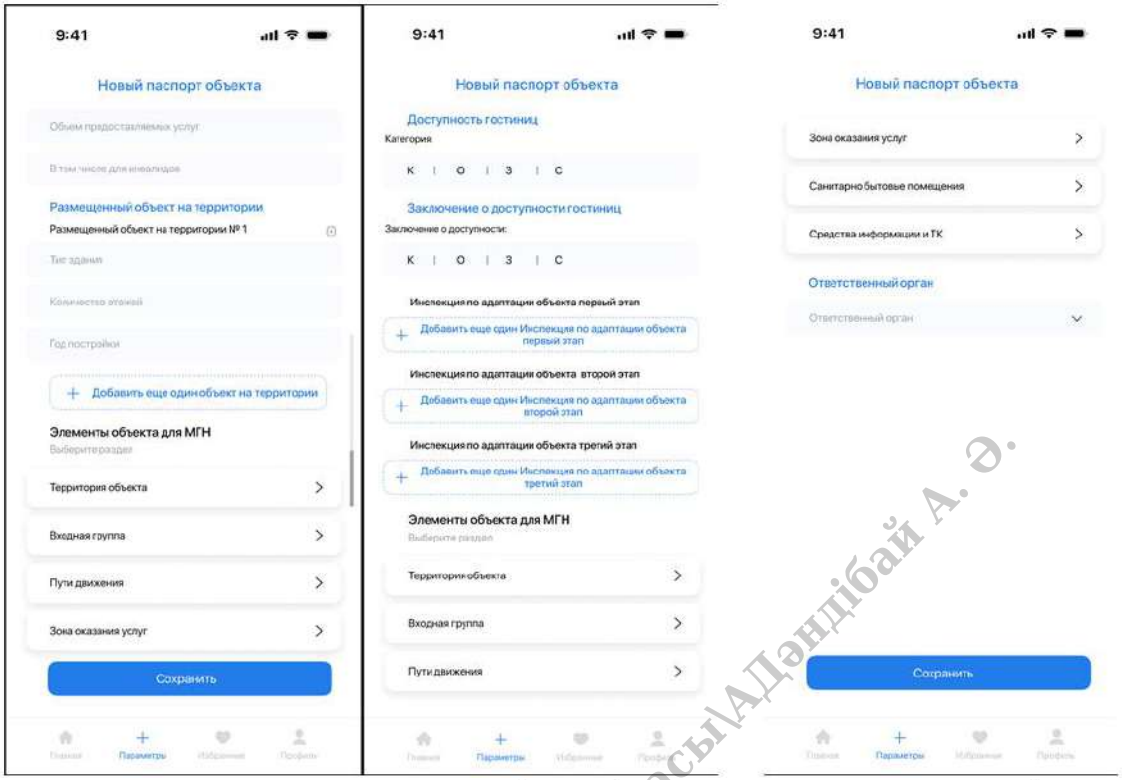
Разработанный модуль:

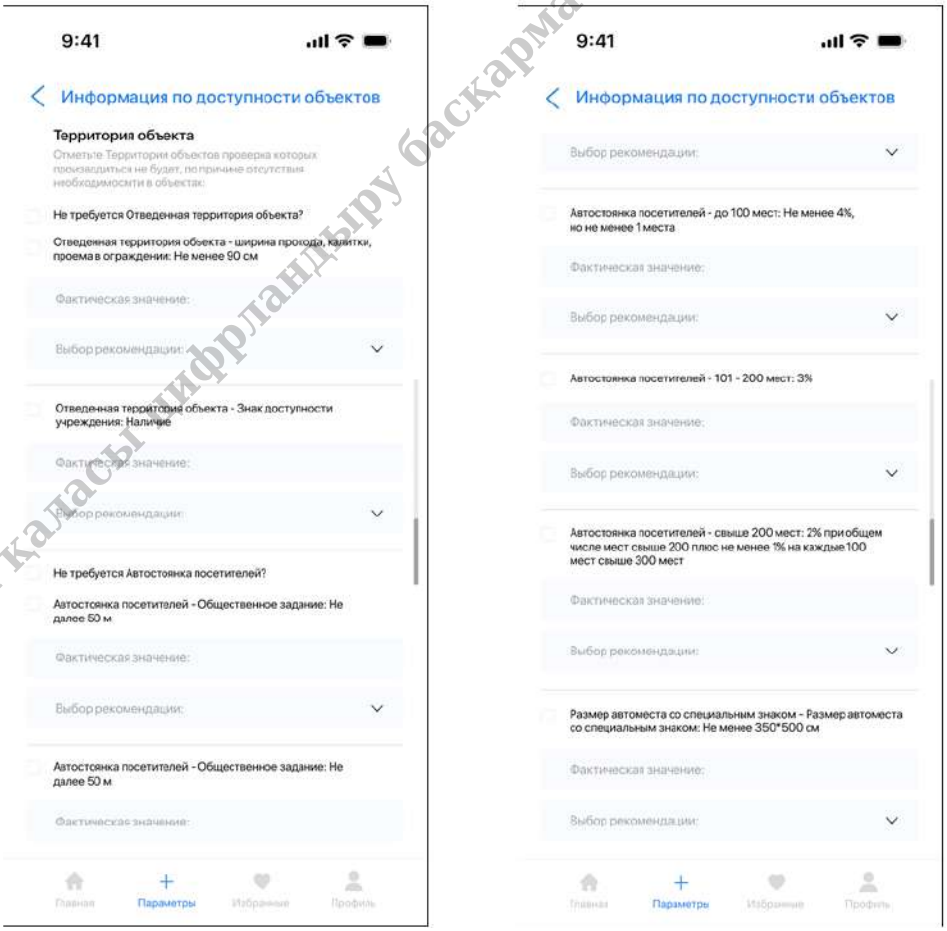
- полностью покрывает государственные требования по паспортизации объектов;
- вмещает сотни параметров и десятки форм, оставаясь удобным и быстрым для пользователя;
- обеспечивает корректную структуру данных для бэкенда;
- снижает трудозатраты инспекторов за счёт автоматизации логики;
- обеспечивает точную фотофиксацию элементов доступности;
- делает процесс создания паспорта прозрачным и неперегруженным.

Этот компонент является ключевым ядром приложения, определяющим весь дальнейший функционал системы.

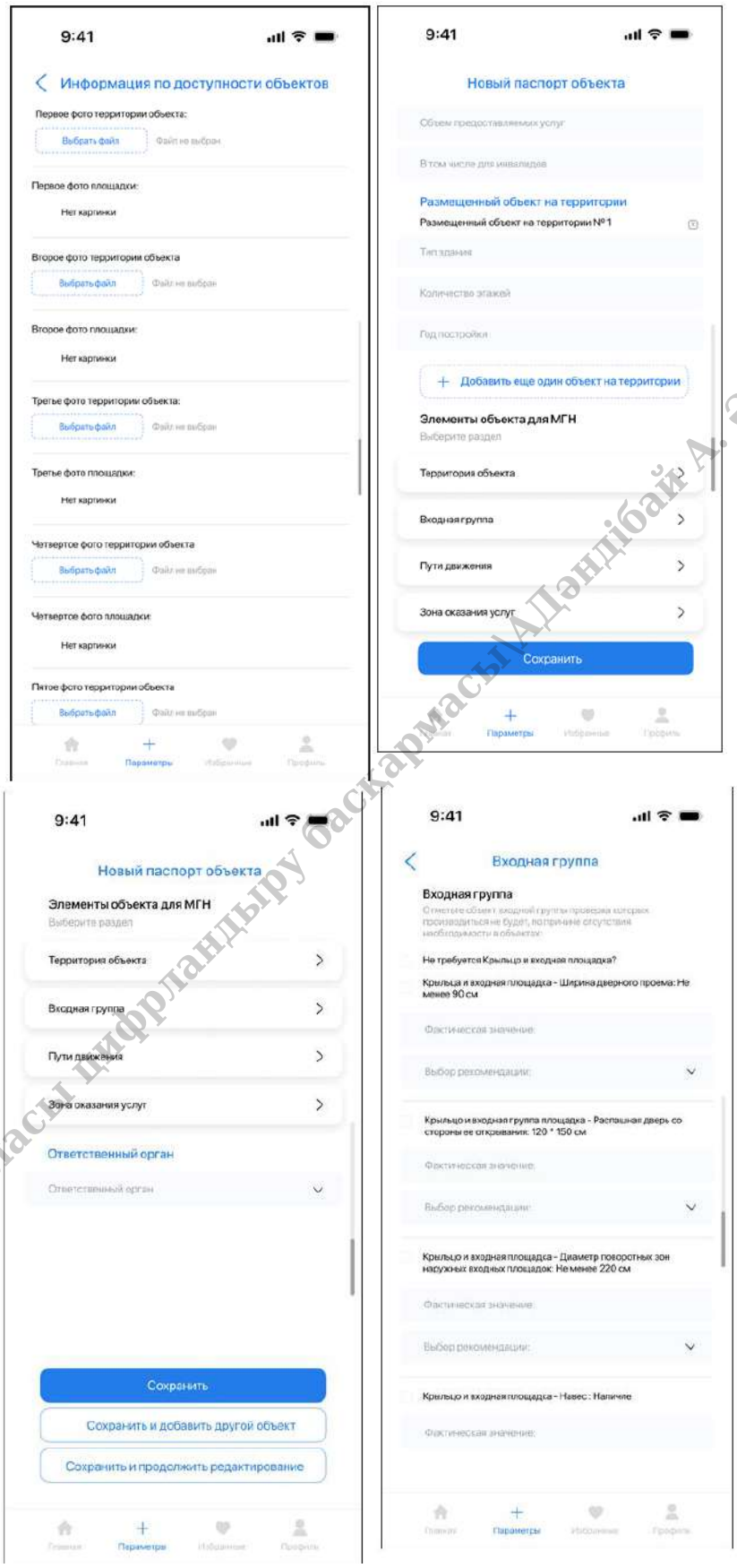



Алматы қаласы цифрландыру басқармасы ҰДәнібай А. Ә.




Алматы қаласы цифрландыру басқармасы\Адәндібай А. Ә.




Алматы қаласы цифрландыру басқармасыАДәндібай А. Ә.

Рисунки – 2.2.1.6.-10. Паспортизация

### 2.2.2. Разработка административной панели для обратной связи от пользователей

В рамках разработки мобильного приложения был разработан и внедрен функционал для формирования обратной связи и рейтинга от пользователей.

Данный функционал предусматривает механизм оценки доступности пользователями и возможность оставления отзывов, где любой зарегистрированный пользователь может:

- поставить оценку удобству и доступности объекта;
- отметить проблемные зоны (например, отсутствие пандуса или неисправный лифт);
- прикрепить фото или комментарий с описанием проблемы;
- предложить улучшения.

Все оценки сохраняются и формируют сводный рейтинг доступности, который отображается на карте. Это создаёт динамическую систему, где пользователи влияют на развитие инфраструктуры и формируют реальную картину инклюзивности города.

***Персонализация маршрутов для пользователей с ЛСИ***

Одной из инновационных функций проекта, является возможность персонализации маршрутов с учётом потребностей конкретного пользователя.

Система анализирует данные о доступных объектах и инфраструктуре, после которого строит маршруты, оптимальные для:

- людей с ограниченной подвижностью (с учётом наличия пандусов, лифтов и подъёмников);
- пользователей с нарушениями зрения (с учётом тактильных плит, светофоров со звуковыми сигналами);
- людей с нарушениями слуха (с учётом наличия визуальных указателей и табло).

Пользователь может выбрать критерии (например, «избегать лестниц», «использовать только доступные тротуары»), а система предложит маршрут с минимальными барьерами.

Таким образом, сервис превращается не только в базу данных, но и в полноценный инструмент навигации и поддержки пользователей с инвалидностью.

***Поиск объектов по категориям доступности***

Также в мобильное приложение внедрен интеллектуальный поиск объектов, адаптированный под разные категории пользователей и уровни доступности.

Пользователи могут фильтровать объекты по различным критериям, такими как:

Алматы қаласы цифрландыру басқармасы Адилбай А. С.

- тип объекта (административные здания, аптеки, магазины, школы, больницы, транспортные узлы и т.д.);
- степень доступности для различных групп ЛСИ (колясочники, слабовидящие, слабослышащие и т.д.);
- наличие инфраструктуры доступности (пандусы, лифты, кнопки вызова, тактильные указатели, понижения бордюров);
- соответствие объектам национальных и международных стандартов инклюзивности.

Поиск реализован через интуитивный интерфейс с подсказками и возможностью географического фильтра — пользователь может искать объекты в своём районе или построить выборку в пределах города.

В будущем предусмотрено внедрение системы умного ранжирования, которая будет предлагать наиболее релевантные объекты на основе истории взаимодействий пользователя.

***Визуализация данных о доступности на карте***

Вся информация, собранная системой, отображается в виде интерактивной карты доступности, предоставляющей пользователям наглядное представление о состоянии городской среды.
Каждый объект имеет свой статус:

- 🟢 Доступен — объект полностью соответствует требованиям доступности;
- 🟡 Частично доступен — часть элементов доступна, но есть барьеры;
- 🔴 Недоступен — объект не адаптирован для ЛСИ.

Карта поддерживает слои визуализации по типам инвалидности, а также отображает маршруты движения с указанием барьеров и точек комфорта.

Пользователи могут увеличивать масштаб, переключать категории и быстро получать карточку объекта с фото, описанием и оценками других пользователей.

Таким образом, визуализация превращается в интерактивный инструмент принятия решений, а не просто в отображение данных.

***Настройка персонализированных маршрутов***

Сервис предоставляет возможность построения персонализированных маршрутов, полностью адаптированных под потребности конкретного пользователя.

Функционал учитывает:

- физические ограничения (например, маршруты без лестниц или крутых спусков);
- особенности передвижения (использование инвалидной коляски, трости, слухового аппарата и т.д.);

Алматы қаласы цифрландыру басқармасы "Дәндібай А. Ә.

- предпочтения пользователя (например, “использовать только проверенные объекты” или “избегать временных барьеров”).

Система анализирует данные о городских объектах и инфраструктуре, используя алгоритмы маршрутизации и геоаналитики.

В результате пользователь получает оптимальный путь, который не только учитывает расстояние, но и гарантирует физическую доступность на всём протяжении маршрута.

Кроме того, при построении маршрута учитываются временные барьеры, такие как:

- ремонтные или строительные работы;
- временно закрытые пандусы или лифты;
- погодные ограничения, влияющие на перемещение (например, гололёд).

Информация обновляется динамически, что делает маршруты максимально точными и актуальными.

***Оставление отзывов и комментариев***

Каждый пользователь может принять участие в оценке качества городской среды, оставив отзыв или комментарий об объекте.

Отзывы позволяют:

- указать на реальные проблемы (например, неисправный лифт, скользкий пандус, плохую навигацию);
- прикрепить фотографии или видео для подтверждения;
- поделиться личным опытом использования объекта;
- предложить конкретные улучшения.

Система отзывов формирует коллективную оценку доступности, где голос каждого пользователя имеет значение.

Эти данные не только отображаются в профиле объекта, но и передаются в агрегированные отчёты для государственных органов и аналитических центров.

***Интеграция с внешними источниками***

Система способна автоматически собирать данные из существующих государственных и общественных карт доступности.

Это обеспечивает наполнение базы проверенной и официальной информацией, а также позволяет объединить в одном месте данные из разных ведомств и организаций.

В рамках интеграции с внешними системами, на тестовом уровне проведена интеграция с ПАК «OpenAlmaty» для обработки информации о проблемах и факта доступности объекта для ЛСИ, которые поступают от пользователей.

Вся зафиксированная информация передается в ПАК «OpenAlmaty» для дальнейшей отработки уполномоченными органами.

Алматы қаласы цифрландыру басқармасы\Әндібай А. Ә.

По факту обработки сообщений и обращений, пользователь приложения получает уведомления с результатами рассмотрения.

***Уведомления о новых и изменённых объектах***

В приложение был внедрен функционал отправки уведомлений, где пользователи могут подписываться на уведомления о новых объектах или изменениях их статуса.

Например:

- добавление нового доступного объекта в районе проживания;
- изменение статуса объекта с “частично доступного” на “доступный”;
- появление временных ограничений, влияющих на маршрут.

Такая система уведомлений помогает пользователям быть в курсе изменений городской инфраструктуры в режиме реального времени.

***Сообщество и обмен опытом***

В мобильное приложение была внедрена отдельная площадка для обсуждений, которая позволяет пользователям обмениваться опытом, задавать вопросы, получать ответы от других пользователей, специалистов, делиться материалами, решать профессиональные задачи, обсуждать сложные ситуации и находить совместные решения.

Данная площадка нужна для обмена опытом, объединения пользователей, быстрого поиска ответов на вопросы, развития новых идей и создания среды поддержки, в которой пользователи могут делиться практическими кейсами и получать полезную обратную связь.

Пользователи могут публиковать свои истории, комментировать опыт других, задавать вопросы и предлагать инициативы.

Это формирует живую экосистему взаимодействия, объединяющую ЛСИ, экспертов и волонтёров в целях развития инклюзивной городской среды.

### 2.2.3. Тестирование пользовательских сценариев

Платформа разработана с учётом реальных сценариев использования:

- ЛСИ получают уведомления о событиях, объектах и маршрутах, которые соответствуют их индивидуальным потребностям.
- При построении маршрута система отображает временные барьеры — например, участки дорог с ремонтными работами или временно закрытые подземные переходы.
- Платформа способна генерировать статистические отчёты и аналитические графики, используемые в научных исследованиях, публикациях и муниципальном планировании.

Таким образом, система не только помогает пользователям передвигаться, но и становится источником данных для принятия управленческих решений и анализа уровня инклюзивности городской среды.

***Результаты и ожидаемый эффект***

Алматы қаласы цифрландыру басқармасы\Дендібай А.О.

Реализация данного функционала принесёт ощутимые социальные и инфраструктурные результаты:

**1.** Повышение удобства передвижения ЛСИ — пользователи смогут планировать маршруты без барьеров и заранее знать о доступных объектах.

**2.** Улучшение городской инфраструктуры — государственные органы и владельцы объектов смогут оперативно получать обратную связь и приоритизировать устранение проблемных зон.

**3.** Формирование инклюзивного цифрового сообщества, где каждый участник способствует созданию доступного и безопасного городского пространства.

Рост информированности населения и стимулирование бизнеса к улучшению стандартов доступности.

### 2.2.4. Публикация приложения в Google Play и App Store

В рамках вывода мобильного приложения Ситуационного центра города Алматы в промышленную эксплуатацию был выполнен полный цикл подготовки и публикации приложения в магазинах Google Play и App Store. Работы включали техническую подготовку билдов, настройку учётных записей разработчика, прохождение процедур валидации и соответствие требованиям обеих платформ.

На стороне Android выполнено формирование релизного APK/AAB-пакета, настройка подписания приложения и загрузка сборки в Google Play Console. Проведена конфигурация карточки приложения, включающая загрузку скриншотов, описания, категорий, возрастного рейтинга, метаданных и настроек конфиденциальности. Приложение успешно прошло автоматическую и ручную модерацию Google.

На стороне iOS выполнена сборка приложения под архитектуру iOS, настройка Provisioning Profiles и сертификатов подписи. Через Xcode и App Store Connect выполнена загрузка IPA-файла, оформление карточки приложения, добавление обязательной информации о политике конфиденциальности, пользовательском соглашении, поддерживаемых устройствах, а также подготовка графических материалов для витрины App Store. Приложение прошло процесс App Review и получило статус «Ready for Sale».

После успешной модерации на обеих платформах выполнены:

- активация публичного доступа для пользователей;
- настройка автоматической доставки обновлений;
- включение аналитики (Firebase / App Store Analytics) и систем краш-мониторинга;
- публикация ссылок на приложение в официальных сервисах Ситуационного центра.

Публикация завершена в установленный срок, приложение доступно широкому кругу пользователей в Google Play и App Store.

Алматы қаласы цифрландыру бескармасы\Дәнгібай А. Ә.

**2.3.Реализация модуля «Мониторинг районов».**

***Цель:***

Развитие модуля мониторинг договоров разработанного в 1 полугодие 2025 года в рамках развития модуля «Мониторинг районов», для реализации дополнительного запроса со стороны Департамента экономических расследований по г. Алматы.

***Задачи:***

4-й квартал:

- Развитие администрации данных под дополнительные параметры
- Разработка Карты для демонстрации информации по строительным работам

***Заказчик:*** Дирекция проектного офиса

### 2.3.1. Развитие администрации данных под дополнительные параметры

Для обеспечения корректной, последовательной и стандартизированной загрузки данных о строительных и ремонтных работах была развернута и детально настроена административная панель, которая служит центральным инструментом для управления всей информацией.

Она позволяет не только вносить и редактировать данные, но и полноценно классифицировать объекты, задавая для каждого из них:

- наименование и подробное текстовое описание;
- категории и подкатегории, отражающие тип и назначение объекта;
- функциональные группы, к которым относится проект;
- параметры программ и дополнительные атрибуты;
- форматы и этапы реализации;
- справочную информацию, включая статусы, районы, ответственные органы, курирующие подразделения и другие структурные элементы.

Административная панель обеспечивает полный цикл подготовки данных перед их публикацией в пользовательском модуле ГИС.

Она поддерживает структурирование, валидацию и унификацию всей вводимой информации, предотвращает дублирование и ошибки, формирует единые стандарты заполнения проектной документации и тем самым гарантирует целостность и согласованность данных в системе.

Алматы қаласы инфрақұрылымды басқармасы/Дәндібай А. Ә.

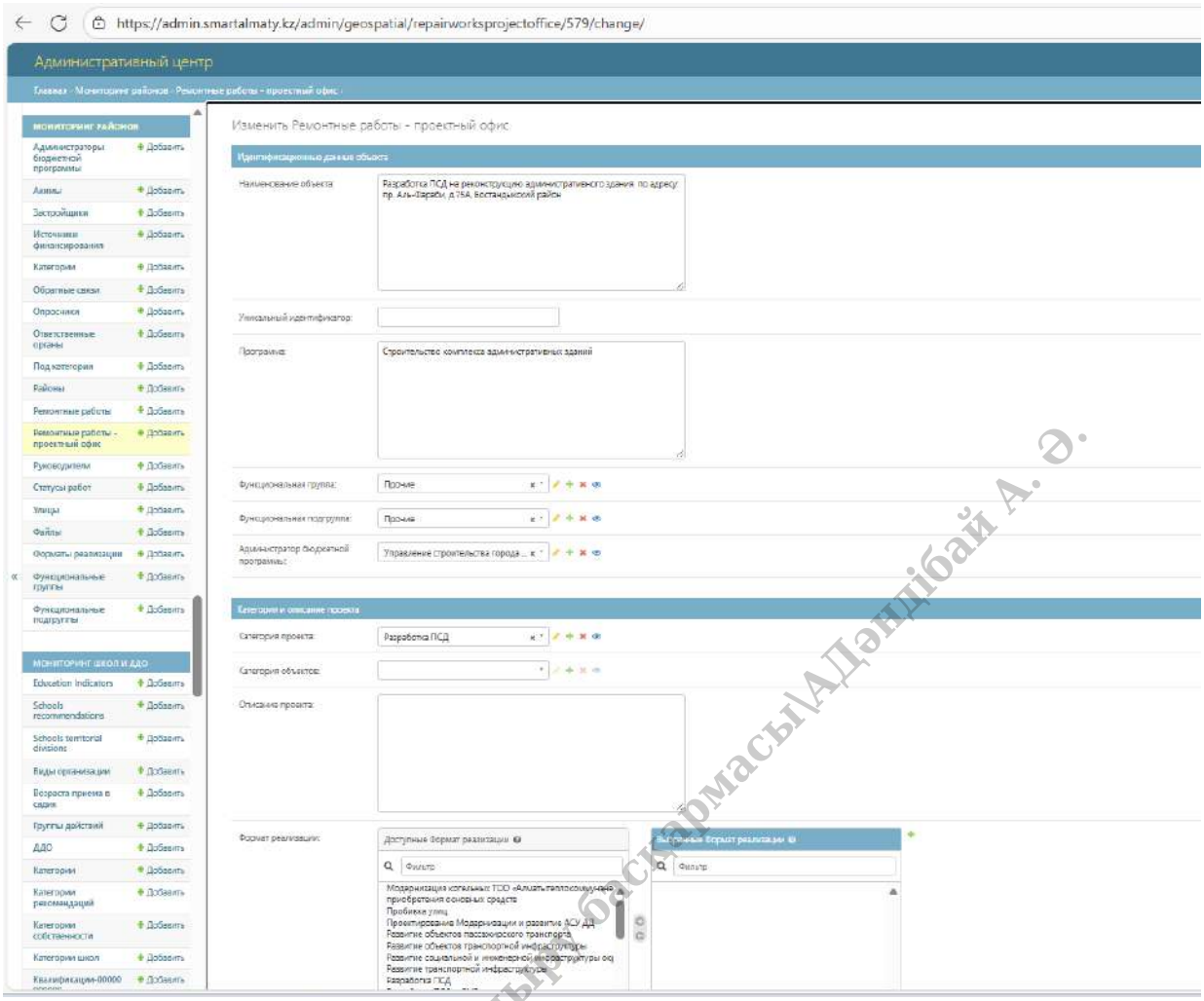



Рисунок - 2.3.1.1. Панель администратора

### 2.3.2. Разработка Карты для демонстрации информации по строительным работам

Модуль «Строительные работы» предназначен для комплексного отображения и контроля хода строительных, ремонтных и благоустроительных работ, проводимых на территории города Алматы. Информация в модуле формируется на основе данных, предоставляемых проектным офисом акимата города Алматы. Система обеспечивает доступ как к планируемым, так и к текущим проектам, позволяя оперативно отслеживать стадии реализации, ответственных исполнителей, сроки, технические параметры и результаты выполнения. https://opendata.smartalmaty.kz/i-construction-works/

**Проводимые работы**

1. Работы по данным. Выполнена загрузка, очистка, нормализация и структурирование данных строительных объектов, включая формирование справочников, связей и геометрических данных для корректного отображения в ГИС-модуле.

2. Работы по backend-части. Реализована серверная логика модуля: разработаны Entity-модели, создано хранилище данных, написаны JPQL-

Алматы қаласы цифрландыру басқармасы\АДәндібай А. Ә.

запросы для выборки объектов и категорий, а также внедрены RESTful API для передачи паспортов и параметров строительных работ.

3. Работы по frontend-части. Создан интерфейс визуализации объектов на ГИС-карте с поддержкой фильтрации, поиска, отображения геометрии и детального просмотра паспорта объекта в пользовательском режиме.

4. Работы по административной панели. Разработана админ-панель для проектного офиса, обеспечивающая ввод, редактирование и ведение паспортов объектов, включая технические характеристики, сроки, участников, документы, фотографии, геометрию и тд.

***1. Загрузка объектов в базу данных SitCenter***

В данном разделе представлена выполненная работа по переносу данных об объектах Бюджета Инвестиционных Программ (БИП) в информационную систему SitCenter. Загрузка включала подготовку исходных данных, формирование справочников, нормализацию записей, создание связей между объектами и обработку географической информации.

***1.1. Подготовка и анализ данных***

Исходным источником являлся Excel-файл «БИПы — все объекты.xlsx», содержащий 582 записи. Для предварительной обработки была создана тестовая таблица bip_objects_damir, в которую загружался весь массив данных для выполнения дальнейшей очистки и валидации.

На этапе анализа проводилась проверка структуры файла, корректности значений, дублирующих записей и единообразия форматов. Данные приводились к стандартному виду, устранялись неточности в наименованиях, датах и числовых значениях.

***1.2. Создание и заполнение справочников***

Для обеспечения структурированного хранения информации в системе были сформированы и заполнены следующие справочники:

- repair_work_functional_group — функциональные группы объектов;
- repair_work_functional_subgroup — функциональные подгруппы;
- repair_work_investition_source — источники финансирования;
- repair_work_realization_format — форматы реализации проектов;
- repair_work_admin_of_budget_program — администраторы бюджетных программ.

Каждый справочник содержит поля id, name_kz, name_ru, что обеспечивает двуязычное представление данных и удобство использования в пользовательских интерфейсах.

Формирование справочников позволило унифицировать исходные данные и обеспечить корректное дальнейшее сопоставление записей при загрузке объектов.

***1.3. Формирование связующей таблицы (many-to-many)***

В целях отражения ситуации, когда один объект БИП может иметь несколько форматов реализации, была создана таблица взаимосвязей repair_works_project_office_realization_format. Заполнение выполнялось с

Алматы қаласы цифрландыру басқармасы/Дәндібай А.Ә.

использованием SQL-скриптов, в которых применялась нормализация данных (TRIM, LOWER) для корректного сопоставления значений из Excel со справочниками. Всего сформировано 582 связи, соответствующие количеству объектов.

### *1.4. Нормализация и корректировка данных*

В тестовой таблице была выполнена комплексная нормализация данных, включающая:

- приведение регистра текстовых значений для обеспечения единообразия;
- унификацию названий статусов и категорий;
- заполнение отсутствующих значений, включая перенос значений общей суммы проекта в поле финансирования по годам.

Например, статус «объявление, после повторного проектирования» был преобразован в унифицированный вид: «Объявление, после повторного проектирования».

Проведённые корректировки обеспечили возможность корректной загрузки данных в основную таблицу и корректную работу связей между справочниками.

### *1.5. Загрузка данных в основную таблицу системы*

После завершения нормализации из тестовой таблицы были перенесены данные в основную таблицу repair_works_project_office, в которую вошли:

- основные сведения об объектах — наименование, программа, сроки реализации, общая стоимость;
- служебные поля — created_at, updated_at, is_blocked, is_deleted;
- ссылки на справочники — категория, подкатегория, район, источник финансирования, администратор бюджетной программы;
- текстовые поля исходных данных, сохранённые для обратной совместимости.

Загрузка прошла успешно, и все 582 объекта были добавлены в систему.

### *1.6. Обработка геоданных*

Работа с геоданными включала два основных этапа:

1. Извлечение координат. Из текстовых описаний объектов автоматически извлекались координаты с использованием регулярных выражений через скрипт *extract_coords_and_geocode.py*.

2. Геокодирование. Для объектов, где координаты отсутствовали, выполнялось определение местоположения с помощью геокодирования на базе OpenStreetMap API.

Полученные координаты конвертировались в геометрический формат и загружались в базу данных при помощи скрипта *insert_in_db_coords.py*. Это обеспечило корректное отображение объектов БИП на карте в модуле SitCenter.

### *1.7. Итоговые результаты*

Выполненные работы позволили завершить полный цикл подготовки

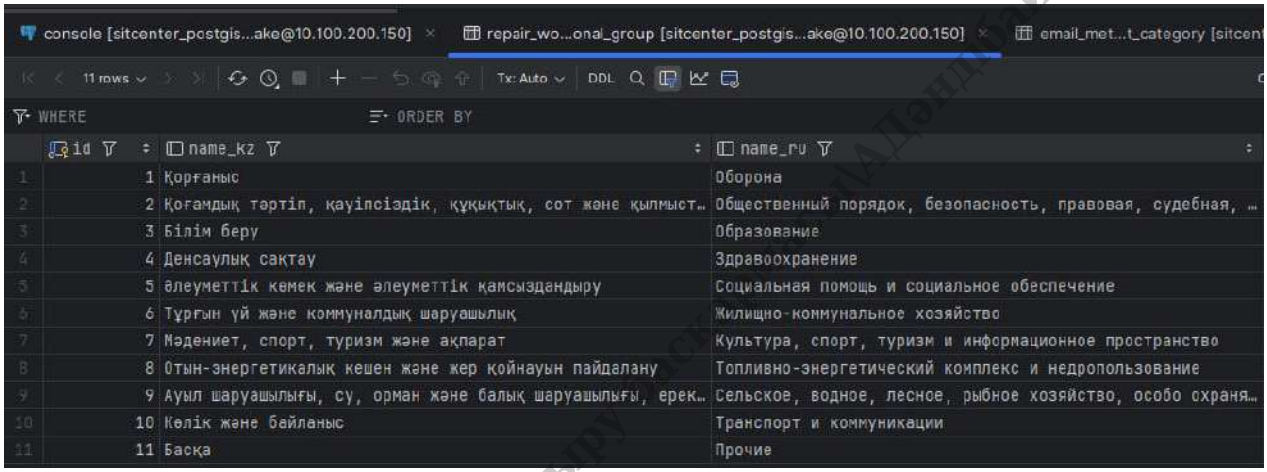
Алматы қаласы цифрландыру басқарласы АДәндібай А. Ә.

данных для модуля «Строительные работы».

Результаты включают:

- успешную загрузку 582 объектов БИП;
- создание и заполнение 5 справочников;
- формирование связующей таблицы форматов реализации;
- нормализацию и очистку данных;
- добавление географических координат и подготовку данных для работы ГИС;
- готовность данных к использованию в дальнейшем — отображение на карте, формирование паспортов объектов и аналитические задачи.

Данные готовы к использованию в системе управления объектами БИП.



| | id | name_kz | name_ru |
|---|---|---|---|
| 1 | 1 | Қорғаныс | Оборона |
| 2 | 2 | Қоғамдық тәртіп, қауіпсіздік, құқықтық, сот және қылмыст… | Общественный порядок, безопасность, правовая, судебная, … |
| 3 | 3 | Білім беру | Образование |
| 4 | 4 | Денсаулық сақтау | Здравоохранение |
| 5 | 5 | әлеуметтік көмек және әлеуметтік қамсыздандыру | Социальная помощь и социальное обеспечение |
| 6 | 6 | Тұрғын үй және коммуналдық шаруашылық | Жилищно-коммунальное хозяйство |
| 7 | 7 | Мәдениет, спорт, туризм және ақпарат | Культура, спорт, туризм и информационное пространство |
| 8 | 8 | Отын-энергетикалық кешен және жер қойнауын пайдалану | Топливно-энергетический комплекс и недропользование |
| 9 | 9 | Ауыл шаруашылығы, су, орман және балық шаруашылығы, ерек… | Сельское, водное, лесное, рыбное хозяйство, особо охраня… |
| 10 | 10 | Көлік және байланыс | Транспорт и коммуникации |
| 11 | 11 | Басқа | Прочие |

Рисунок – 2.3.1.2. Справочник категорий объектов БИП с двуязычными названиями

| | id | name_kz | name_ru |
|---|---|---|---|
| 1 | 15 | Әлеуметтік қамсыздандыру | Социальное обеспечение |
| 2 | 8 | Бастауыш, негізгі және жалпы орта білім беру | Начальное, основное среднее и общее среднее образование |
| 3 | 2 | Қоғамдық даму | Благоустройство населенных пунктов |
| 4 | 10 | Қоғамдық денсаулық сақтау | Охрана здоровья населения |
| 5 | 6 | Коммуналдық қызметтер | Коммунальное хозяйство |
| 6 | 18 | Отын-энергетика | Топливо и энергетика |
| 7 | 1 | Автомобиль көлігі | Автомобильный транспорт |
| 8 | 13 | Басқа | Прочие |
| 9 | 16 | Спорт | Спорт |
| 10 | 11 | Қоршаған ортаны қорғау | Охрана окружающей среды |
| 11 | 4 | Теміржол көлігі | Железнодорожный транспорт |
| 12 | 17 | Техникалық, кәсіптік және жоғары білім беру | Техническое и профессиональное, послесреднее образование |
| 13 | 12 | Құқық қорғау | Правоохранительная деятельность |
| 14 | 9 | Төтенше жағдайларды басқару | Организация работы по чрезвычайным ситуациям |
| 15 | 7 | Мәдениет | Культура |
| 16 | 3 | Мектепке дейінгі білім беру және оқыту | Дошкольное воспитание и обучение |
| 17 | 5 | Тұрғын үй | Жилищное хозяйство |
| 18 | 14 | Басқа көлік және коммуникация қызметтері | Прочие услуги в сфере транспорта и коммуникаций |

Рисунок – 2.3.1.3. Справочник функциональных подгрупп объектов БИП с двуязычными названиями

| | id | name_kz | name_ru |
|---|---|---|---|
| 1 | 1 | Жергілікті бюджет | Местный бюджет |
| 2 | 2 | Ұлттық қор | Национальный фонд |
| 3 | 3 | Республикалық бюджет | Республиканский бюджет |
| 4 | 4 | Бірлесіп қаржыландыру | Софинансирование |

Рисунок – 2.3.1.4. Справочник источников
финансирования объектов БИП с двуязычными названиями

```sql
update bip_objects_damir
set этап_реализации___статус = 'Объявление, после повторного проектирования'
where этап_реализации___статус in ('объявление, после повторного проектирования');
```

Рисунок – 2.3.1.5. SQL-запрос для
нормализации данных статусов реализации в тестовой таблице

```sql
insert into repair_works_project_office (object_name,
                                         --functional_group_id,
                                         --functional_sub_group_id,
                                         --admin_of_budget_program_id,
                                         program,
                                         start_date,
                                         end_date,
                                         --category_id,
                                         total_price,created_at,updated_at, is_blocked,
                                         is_deleted,category_id,
                                         contractor_id,district_id,
                                         investition_source_id,status_id,sub_category_id,
                                         func_group,subfunc_group,investition_source,admin_of_budget_program,
                                         realization_format)

select наименование_проекта,программа,now(),now(), бюджет, now(), now(),false, false,1,1,1,1,1,5,
       функциональная_группа,функциональная_подгруппа,источник,администратор_бюджетной_программ,формат_реализации
from bip_objects_damir order by "№";
```

Рисунок – 2.3.1.6. SQL-запрос для загрузки данных из тестовой таблицы в
основную таблицу объектов БИП

```python
def extract_coords(text: str):
    if not isinstance(text, str):
        return None, None

    m = coord_pattern.search(text)
    if not m:
        return None, None

    v1 = float(m.group(1))
    v2 = float(m.group(2))

    # Определяем, что lat в диапазоне [-90..90]
    if -90 <= v1 <= 90 and -180 <= v2 <= 180:
        return v1, v2
    if -90 <= v2 <= 90 and -180 <= v1 <= 180:
        return v2, v1

    return None, None
```

Рисунок – 2.3.1.7. Извлечение координат
из текста через регулярные выражения

Алматы қаласы цифрлаңдыру ... масы\АДәндібай А

```python
def geocode_address(addr: str):
    if not addr:
        return None, None

    params = {
        "q": addr,
        "format": "json",
        "limit": 1,
    }

    url = "https://nominatim.openstreetmap.org/search?" + urllib.parse.urlencode(params)

    try:
        r = requests.get(
            url,
            headers={"User-Agent": "Mozilla/5.0"},
            timeout=15  # устойчивый таймаут
        )
        r.raise_for_status()
        data = r.json()

        if not data:
            return None, None

        lat = float(data[0]["lat"])
        lon = float(data[0]["lon"])
        return lat, lon

    except Exception as e:
        print(f"Ошибка геокодера для '{addr}': {e}")
        return None, None
```

Рисунок – 2.3.1.8. Геокодирование адресов через OpenStreetMap API

```python
    for _, row in df.iterrows():
        obj_id = row.get("id")
        wkt_point = row.get("wkt_point")

        if pd.isna(wkt_point):
            continue

        # получаем hex WKB
        wkb_hex = wkt_to_wkb_hex(wkt_point)

        sql = f"""
            UPDATE repair_works_project_office
            SET marker = ST_SetSRID('{wkb_hex}'::geometry, 4326)
            WHERE id = %s
              AND marker IS NULL;
        """

        cur.execute(sql, (obj_id,))
        updates += cur.rowcount
```

Рисунок – 2.3.1.9. Обновление геоданных в базе данных

В 4-кварталe были выполнены работы по модулю «Мониторинг районов» в части добавления информации о строительных работах. В рамках данной задачи на стороне backend были выполнены следующие работы:

- Разработка Entity gis_repair_works_project_office таблицы для хранения ремонтных работ;
- Разработка пейлод интерфейса для гис объектов ремонтных работ;
- Разработка пейлод интерфейса для паспорта объекта ремонтных работ;
- Разработка JPQL запроса в БД для передачи паспорта объекта ремонтных работ; (Рисунок – 2.3.1.10)
- Разработка JPQL запроса в БД для передачи ГИС объектов ремонтных работ; (Рисунок - Рисунок – 2.3.1.11)
- Разработка JPQL запроса в БД для передачи справочника категории ремонтных работ; (Рисунок – 2.3.1.12)

Алматы қаласы ИИД Қарасай аудандық сотының Апшибай А. Ә.

- Разработка RESTful API для передачи паспорта объектов ремонтных работ; (Рисунок – 2.3.1.13)
- Разработка RESTful API для передачи ГИС объектов ремонтных работ. (Рисунок – 2.3.1.14)

***Разработка серверной сущности (Entity) для хранения данных ремонтных работ***

В системе была создана сущность **gis_repair_works_project_office**, представляющая собой серверный объект, описывающий структуру таблицы в базе данных. Данная сущность объединяет все ключевые характеристики ремонтных работ: основные атрибуты объекта, категорию, подкатегорию, статус, район, источник финансирования, формат реализации, геометрию объекта, сроки и служебную информацию.

Разработка сущности обеспечила единый формат данных, используемый как в интерфейсе, так и в API.

***Разработка payload-структуры для ГИС-объектов ремонтных работ***

Для передачи данных в ГИС-слой была разработана отдельная структура **GIS-payload**, включающая:

- идентификатор объекта;
- наименование;
- краткий адрес;
- категорию и подкатегорию;
- геометрию (Point, LineString, Polygon или MultiPolygon);
- статус и формат реализации;
- сроки проведения работ;
- координатный маркер для отображения на карте.

Payload оптимизирован под визуализацию в интерфейсе ГИС и обеспечивает корректное отображение объектов при разных масштабах карты.

***Payload для паспорта объекта ремонтных работ***

Для формирования паспорта объекта разработана отдельная DTO-структура, включающая:

- основную информацию об объекте;
- описание проекта;
- данные по финансированию;
- технические характеристики;
- участников и ответственных лиц;
- паспорта геометрии;
- актуальную стадию реализации;
- причину отклонений и показатели исполнения.

Backend-часть обеспечивает автоматическое формирование паспорта на основании данных, хранящихся в базе, и передачу его в интерфейс по REST-API.

***Разработка JPQL-запросов для выборки данных о ремонтных работах***

Алматы қаласы цифрландыру басқармасы\АЛендібай А. Э.

Для получения детализированной информации о каждом объекте были разработаны сложные JPQL-запросы, включающие множественные LEFT JOIN с таблицами-справочниками.

***Запрос для получения паспорта объекта***

JPQL-запрос объединяет данные о районе, категории, подкатегории, статусе, источнике финансирования, формате реализации, административной группе, координатах и другой служебной информации.

```
SELECT
rw
FROM GisRepairWorkPO rw
LEFT JOIN rw.district
LEFT JOIN rw.category
LEFT JOIN rw.subCategory
LEFT JOIN rw.status
LEFT JOIN rw.contractor
LEFT JOIN rw.budgetProgram
LEFT JOIN rw.dlStatus
LEFT JOIN rw.status
LEFT JOIN rw.functionalGroup
LEFT JOIN rw.functionalSubGroup
LEFT JOIN rw.investSource
 WHERE rw.id = :id AND rw.isDeleted = FALSE
```

Рисунок – 2.3.1.10. JPQL запрос в БД для передачи паспорта

***Запрос для передачи ГИС-объектов***

Запрос формирует сокращённый набор атрибутов для работы в геоинформационном модуле, включая геометрию, маркер и фильтры по району, категории и статусу.

```
SELECT
rw.id as id,
rw.address as address,
rw.name as name,
rw.district as district,
rw.category as category,
rw.subCategory as subCategory,
rw.status as status,
rw.startDate as startDate,
rw.endDate as endDate,
rw.marker as marker,
rw.geom as geom
FROM GisRepairWorkPO rw
LEFT JOIN rw.district
LEFT JOIN rw.category
LEFT JOIN rw.subCategory
LEFT JOIN rw.status
LEFT JOIN rw.contractor
LEFT JOIN rw.budgetProgram
LEFT JOIN rw.dlStatus
LEFT JOIN rw.status
LEFT JOIN rw.functionalGroup
LEFT JOIN rw.functionalSubGroup
LEFT JOIN rw.investSource
WHERE rw.isDeleted = FALSE AND
(:#{#filter.distId} IS NULL OR rw.district.id = :#{#filter.distId}) AND
(:#{#filter.subCatId} IS NULL OR rw.subCategory.id = :#{#filter.subCatId}) AND
(:#{#filter.statusId} IS NULL OR rw.status.id = :#{#filter.statusId})
```

Рисунок – 2.3.1.11. JPQL запрос в БД для передачи ГИС объектов

Алматы қаласының қоғамдық қауіпсіздігі департаменті бойынша қарастырылады\Адәндібай А. Ә.

***Запрос для передачи справочника категорий***

Запрос формирует список категорий и подкатегорий с подсчётом количества объектов в каждой категории с учётом параметров фильтрации.

```
SELECT
rsc.id as id,
rsc.name as name,
rsc.icon as icon,
rsc.image as image,
rsc.category as category,
(SELECT COALESCE(COUNT(rw.id), 0)
FROM GisRepairWorkPO rw
LEFT JOIN rw.district
LEFT JOIN rw.category
LEFT JOIN rw.subCategory
LEFT JOIN rw.status
LEFT JOIN rw.contractor
LEFT JOIN rw.budgetProgram
LEFT JOIN rw.dlStatus
LEFT JOIN rw.status
LEFT JOIN rw.functionalGroup
LEFT JOIN rw.functionalSubGroup
LEFT JOIN rw.investSource
WHERE rw.isDeleted = FALSE AND
(:#{#filter.distId} IS NULL OR rw.district.id = :#{#filter.distId}) AND
(rw.subCategory.id = rsc.id) AND
(:#{#filter.statusId} IS NULL OR rw.status.id = :#{#filter.statusId})
) as amount
FROM RepairWorkSubCategory rsc
LEFT JOIN rsc.category
WHERE rsc.isDeleted = FALSE
ORDER BY rsc.category.id, rsc.id
```

Рисунок – 2.3.1.12. JPQL запрос в БД
для передачи справочника категории

***Разработка RESTful API для паспорта и ГИС-объектов***

Были разработаны методы REST API, обеспечивающие доступ к данным ремонтных работ извне.

**Метод GET /repair-work/po/{id} — паспорт объекта**

Метод возвращает полный паспорт ремонтного объекта, включая:

- наименование;
- адрес;
- категорию;
- субкатегорию;
- статус;
- геометрию (Point/Polygon/MultiPolygon);
- сроки проекта;
- описание;
- финансирование;
- ответственных лиц;
- фотографии;
- приложенные документы.

Алматы қаласы цифрландыру басқармасы АДәндібай А. Ә.

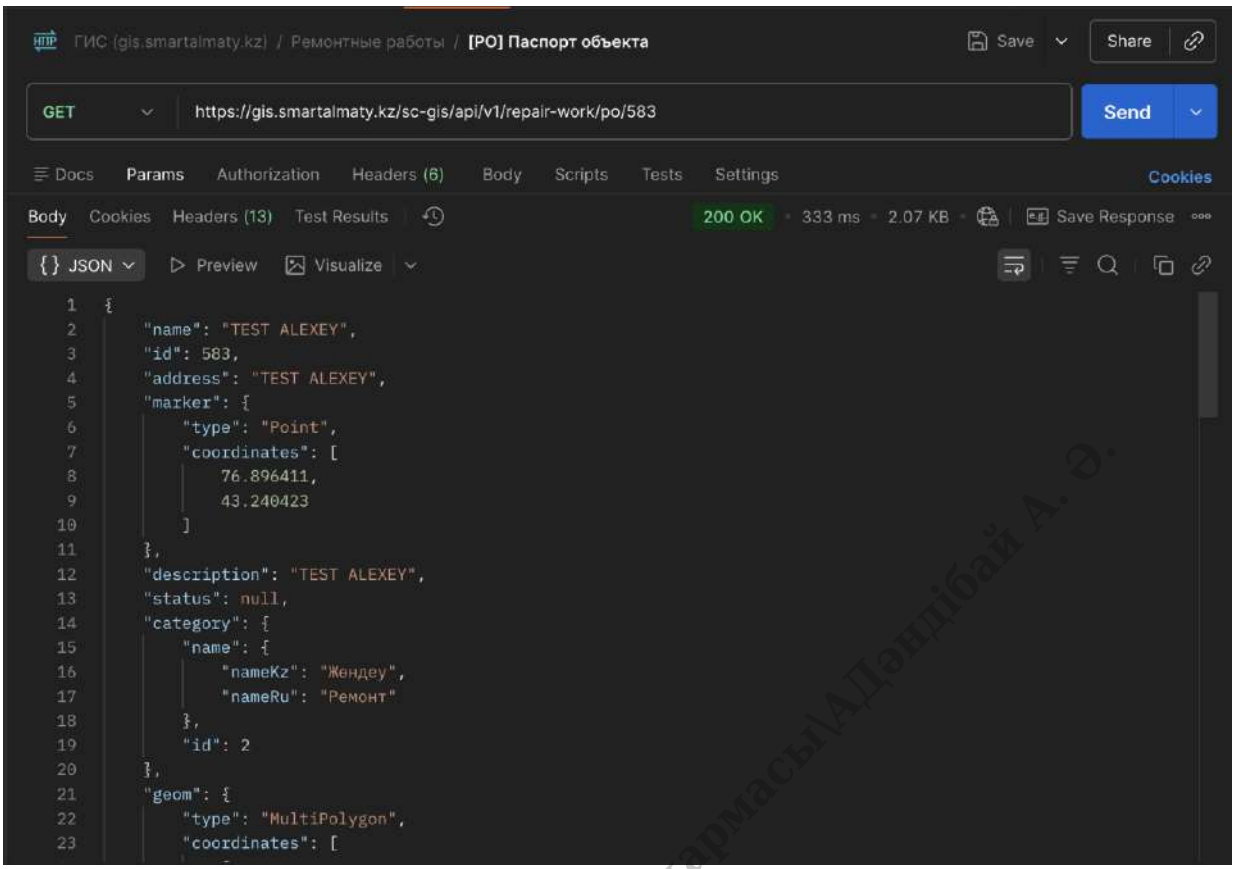



Рисунок – 2.3.1.13. RESTful API для передачи паспорта объектов

***Метод POST /repair-work/po — список объектов для ГИС***

Метод возвращает массив объектов с минимальным набором данных для отображения на карте:

- id;
- имя;
- подкатегория;
- иконка категории;
- геометрия (если имеется);
- маркер;
- сроки реализации.

Алматы қаласы цифрландыру басқармасы\Әдәндібай А. Ә.

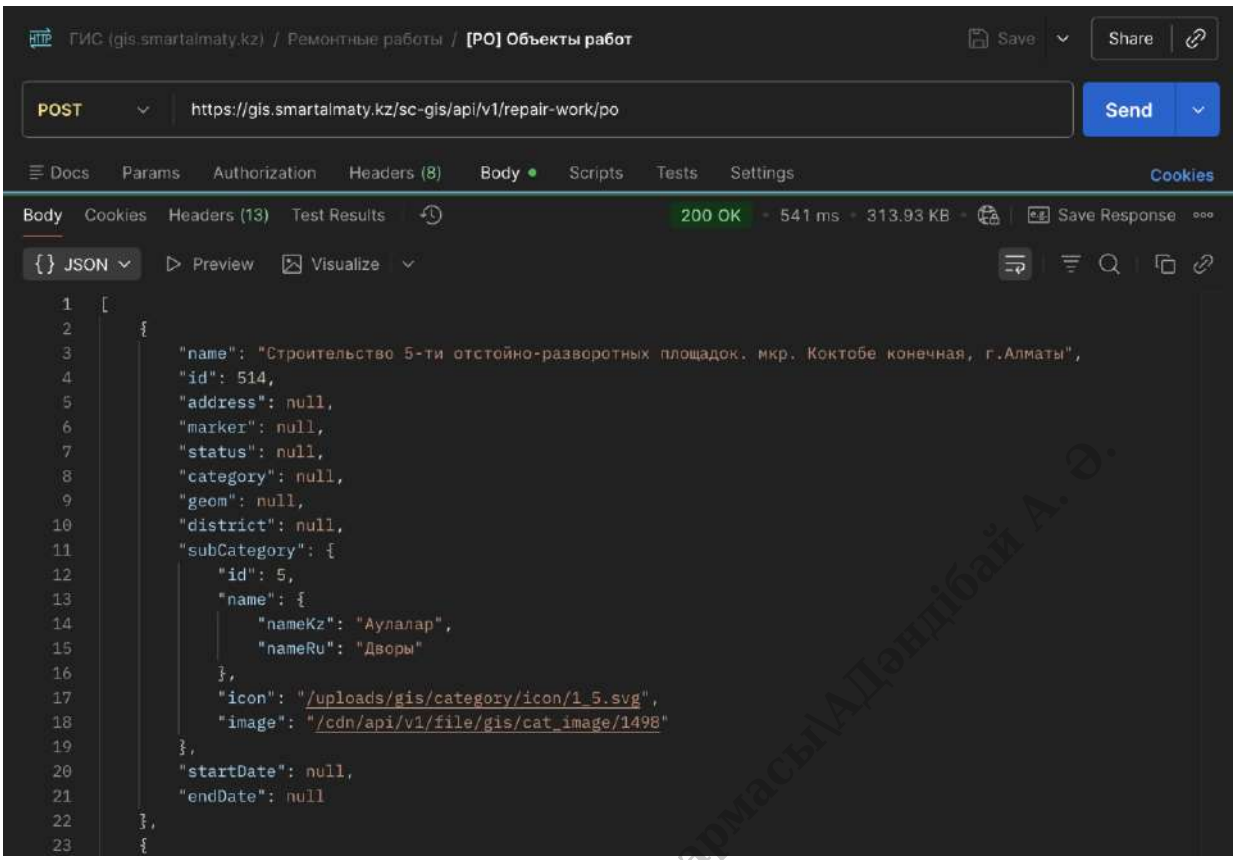



Рисунок – 2.3.1.14. RESTful API для передачи ГИС объектов

***Результаты backend-разработки***

В результате выполненной работы backend-часть модуля была полностью подготовлена для взаимодействия с пользовательскими интерфейсами и ГИС-слоем. Реализованы:

- единая серверная сущность для хранения данных ремонтных работ;
- специализированные payload-структуры;
- три ключевых JPQL-запроса;
- полнофункциональное REST-API;
- поддержка фильтрации, пагинации и выборки данных в зависимости от пользовательского интерфейса;
- корректная передача геоданных и маркеров на карту.

Backend-модуль полностью готов к эксплуатации и интегрирован в систему SitCenter.

На стороне backend были проведены работы по визуализации модуля.

Алматы қаласы инфрақұрылымды басқармасы\АДәндібай А. Ә.

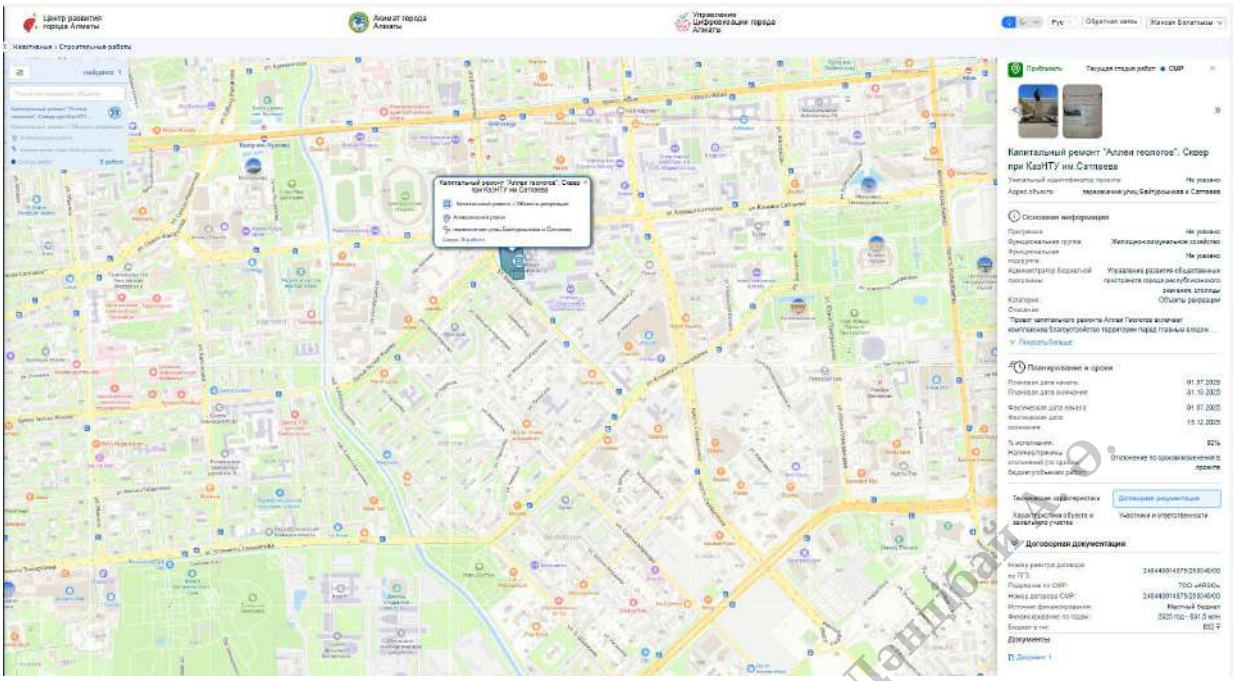


Рисунок – 2.3.1.15. Карта ремонтных работ

***Левая фильтрационная панель***

Слева расположены инструменты поиска и фильтрации объектов:

1. Поле поиска по названию объекта;
2. Фильтр региона: г. Алматы;
3. Фильтр статуса работ;
4. Фильтр исполнителей;
5. Фильтр периода: Текущий год / Выбор года.
6. Категории работ с указанием количества объектов в каждой:
    - Благоустройство;
    - Ремонт;
    - Сейсмоусиление;
    - Строительство;
    - Реконструкция;
    - Капитальный ремонт.

Эти элементы позволяют пользователю быстро находить нужные объекты на карте.

***Центральная ГИС-карта***

Основная часть экрана занята интерактивной картой, на которой отображаются:

- геолокация объектов строительных работ;
- всплывающее окно с краткой информацией об объекте при наведении:
    - название объекта;
    - локация;
    - вид работ;
    - категория работ.

Алматы қаласы цифрландыру басқармасы АДәндібай А.Ә.

Карта позволяет визуально оценивать расположение и плотность распределения объектов.

***Правая информационная панель — Паспорт объекта***

При выборе объекта система раскрывает подробную информацию:

*Заголовок и медиа*

- отображение фотографий объекта;
- стадия выполнения

*Основная информация*

- название
- уникальный идентификатор проекта
- программа
- функциональная группа
- функциональная подгруппа
- администратор бюджетной программы
- описание проекта

*Планирование и сроки*

- плановое начало
- плановое окончание
- фактическое начало
- фактическое окончание
- процент исполнения

*Стадия и отклонения*

- текущая стадия: СМР
- информация об отклонениях — не указана

*Технические характеристики*

- открывается отдельным разделом

*Договорная документация*

- номер реестра договора
- подрядчик по СМР
- номер договора СМР
- источник финансирования
- финансирование по годам
- бюджет

*Документы*

- список прикреплённых документов (доступен для загрузки)

*Технические характеристики объекта*

*Мощность / производительность / пропускная способность объекта*

*Заключение экспертизы (№, дата)*

*Технические условия по подключению к инженерным сетям*

*Характеристики объекта и земельного участка*

- Категория сложности
- Категория сейсмостойкости

Алматы қаласының цифрландыру басқармасы\АДәндібай А. Ә.

- Категория взрывопожароопасности
- Кадастровый номер земельного участка

*Участники и ответственности*
*Проектировщик / Авторский надзор*
*Технический надзор*
*Ответственный за управление проектом*
*Ответственный за эксплуатацию*
*ФИО ответственного от Управления строительства (УСт)*
*ФИО ответственного от Акимата города Алматы*
*ФИО ответственного по мониторингу проекта от ЦРА*

### 2.4.Разработка модуля «Мониторинг сейсмоустойчивости городских объектов»

***Цель:***

Разработать систему визуализации и мониторинга результатов паспортизации городских объектов с точки зрения их сейсмоустойчивости для повышения готовности к чрезвычайным ситуациям.

***Задачи:***

4-й квартал:

- Разработка системы уведомлений для мониторинга изменений в статусе объектов.
- Рекомендации по сейсмоустойчивости для усиления уязвимых объектов.

***Источник данных:*** Управление городского планирования и урбанистики.
***Заказчик:*** Управление сейсмики города Алматы.

### 2.4.1. Разработка системы уведомлений для мониторинга изменений в статусе объектов

В рамках работ 4-квартала в части развития модуля «Мониторинг сейсмоустойчивости городских объектов» на стороне backend были проведены следующие работы:

Работы по обновлению данных по готовности городской инфраструктуры;

Доработка API – были проведены работы по добавлению новых эднпоинтов для расширения возможностей фильтрации данных;

В модуль были добавлены новые параметры для поиска и фильтрации данных.

Алматы қаласы цифрлаңыру басқармасы\АДаулбай А. Э.

```python
def get_mvt_tile(self, z: int, x: int, y: int, gri_class=None, district=None, density_level=None):
    max_val = 2**z - 1
    if x < 0 or y < 0 or x > max_val or y > max_val:
        return None

    filters = []
    params = [z, x, y]

    if gri_class:
        op = ' = ANY(%s)' if isinstance(gri_class, (list, tuple)) else ' = %s'
        filters.append(f't."GRI_class"{op}')
        params.append(gri_class)

    # Логика для district
    if district:
        op = ' = ANY(%s)' if isinstance(district, (list, tuple)) else ' = %s'
        filters.append(f"t.district{op}")
        params.append(district)

    # --- НОВЫЙ БЛОК ДЛЯ DENSITY_LEVEL ---
    if density_level:
        op = ' = ANY(%s)' if isinstance(density_level, (list, tuple)) else ' = %s'
        filters.append(f"t.density_level{op}")
        params.append(density_level)

    where_clause = "AND " + " AND ".join(filters) if filters else ""

    sql = f"""
        WITH bounds AS (
            SELECT ST_Transform(ST_TileEnvelope(%s, %s, %s), 4326) AS geom
        )
        SELECT ST_AsMVT(tile, 'geo_risk', 4096, 'geom') AS mvt
```

```python
class AddressBuildingsRiskService(BasePostGISService):
    def get_mvt_tile(self, z: int, x: int, y: int, filters_dict):
        max_val = 2**z - 1
        if x < 0 or y < 0 or x > max_val or y > max_val:
            return None

        FILTER_CONFIG = {
            'district': 'district',
            'seismic_eval': 'seismic_eval',
            'is_emergency_building': 'is_emergency_building',
            'is_passport': 'is_passport',
            'cluster': 'cluster',
            'group': '"group"',  # ВАЖНО: "group" - ключевое слово SQL, нужно экранировать
            'is_highrise_in_poly': 'is_highrise_in_poly',
            'measure_category': 'measure_category',
            'priority': 'priority',
            'op_order': 'op_order',
            'cluster_rank': 'cluster_rank',
        }

        filters = []
        params = [z, x, y]

        for key, value in filters_dict.items():
            if key in FILTER_CONFIG and value is not None:
                column_name = FILTER_CONFIG[key]

                # Для списков (например, ?district=Район1,Район2)
                if isinstance(value, list):
                    filters.append(f't.{column_name} = ANY(%s)')
                    params.append(value)
                # Для одиночных значений (например, ?is_passport=true)
                else:
                    filters.append(f't.{column_name} = %s')
                    params.append(value)
```

Рисунок – 2.3.2.1. Обновленный API модуля «Мониторинг сейсмоустойчивости городских объектов»

В рамках развития модуля «Мониторинг сейсмоустойчивости городских объектов» была разработана автоматизированная система уведомлений, обеспечивающая отслеживание изменений в статусах обследуемых зданий и своевременное информирование профильных подразделений.

На основе единого геослоя и атрибутивных данных модуля, содержащего информацию о результатах обследования объектов (категория сейсмостойкости, дата проверки, наличие нарушений, уровень риска), был реализован механизм регулярного контроля изменений. Система выполняет периодическое сравнение текущих данных с предыдущими версиями записей, используя идентификаторы

Алматы қаласы Цифрландыру басқармасы\Дәндібай А. Ә.

объектов и ключевые атрибуты состояния. При выявлении несоответствий или обновлений статусов формируется запись об изменении, включающая дату, тип изменения, предыдущий и новый показатели.

Для уведомления ответственных специалистов разработан инструмент автоматической генерации отчётов в формате Excel. Отчёт формируется на основе агрегированных данных по выявленным изменениям за отчётный период и содержит структурированные таблицы с перечнем объектов, изменившими свой статус, а также вспомогательную информацию: адрес, уникальный ID, классификацию изменения и параметры риска.

После формирования файл Excel проходит процедуру автоматической загрузки в почтовый модуль, где с использованием SMTP-клиента выполняется отправка уведомлений на заранее заданный список адресатов. Настроена система расписаний, обеспечивающая ежедневный или недельный режим рассылок, а также возможность ручного запуска формирования отчётов.

Техническая реализация включает:

- модуль сравнения данных с хранением предыдущих версий записей;
- механизм триггеров и условий выявления изменений;
- генератор Excel-файлов на базе шаблонов с динамическим заполнением;
- интеграцию с почтовым шлюзом для автоматизированной рассылки;
- логирование всех операций (выявление изменений, формирование отчёта, отправка писем) для контроля корректности работы.

### 2.4.2. Рекомендации по сейсмоустойчивости для усиления уязвимых объектов

В рамках разработки рекомендации по сейсмоустойчивости для усиления уязвимых объектов, была проработана методология, которая содержит рекомендации, предназначенные для:

- классификации зданий по типу необходимого сейсмического вмешательства;
- количественной оценки выгоды (снижения риска) и стоимости мероприятий;
- формирования приоритетов для реализации (какие здания делать в первую очередь);
- группировки зданий в операционные кластеры, чтобы снизить затраты на выезды и обследования;
- визуализации приоритетов на карте города в виде интерактивного слоя.

Таким образом, рекомендации служат связующим звеном между индексами риска (SRI/GRI) и конкретным планом действий – какие объекты нужно паспортизировать, какие объекты усиливать, и на каких объектах нужно осуществлять реконструкцию или снос с новым строительством.

Алматы қаласы Цифрландыру басқармасы АДәндібай Ә.

Для разработки данной методологии были использованы следующие исходные данные и ключевые параметры:

*Входные параметры:*

- геометрия зданий;
- сейсмический риск *SRI (0–1)*;
- индекс грунтового/геологического риска *GRI (0–1)*;
- компоненты индекса риска *(H, V, при наличии)*;
- характеристики экспозиции: площадь здания, количество этажей, общая площадь (gfa_m2);
- флаги состояния:
  - признак аварийности здания;
  - наличие/отсутствие сейсмического паспорта;
  - факт проведения сейсмической оценки (*seismic_eval*).

*Дополнительно задаются параметры:*

- ориентировочные удельные стоимости:
  - за демонтаж/реконструкцию 1 м²;
  - за усиление 1 м²;
  - за паспортизацию объекта;
- стоимость «дорожного» плеча (затраты на выезд бригады в расчёте на километр);
- целевой уровень *SRI (TARGET_SRI)*, к которому стремится усиление;
- вес неопределённости для паспортизации (*UNCERTAINTY_WEIGHT*).

***Классификация зданий по типам мероприятий***

Для каждого здания определяется категория меры (measure_category):

- Reconstruction/Demolition — реконструкция или снос:
  - аварийные здания (*is_emergency_building = True*);
  - крайне высокий риск и/или невозможность эффективного усиления.
- Passportization — паспортизация:
  - у зданий отсутствует сейсмический паспорт;
  - при этом они не классифицированы как аварийные.
- Strengthening — усиление:
  - у зданий есть паспорт, но не проведена детальная сейсмическая оценка;
  - здания требуют инженерного вмешательства для доведения SRI до целевого уровня.
- Stable case — стабильные объекты (дефолтная категория):
  - при отсутствии признаков аварийности и необходимости мер в текущем цикле.

Эта классификация задаёт первичный тип мероприятия, а дальнейший алгоритм оценивает, в каком порядке и насколько целесообразно реализовывать

Алматы қаласының цифрлық даму басқармасы\АДәндібай А. Ә.

эти меры.

***Оценка риска, выгоды и стоимости мероприятий***

Для каждой категории мероприятий строится свой профиль.

Для всех зданий рассчитывается показатель *risk_now*, отражающий текущий сейсмический риск с учётом грунтовых условий:

*риск = SRI × (0,5 + 0,5 × GRI)*

Таким образом, здания в более опасных грунтовых условиях получают повышенный вес риска даже при одинаковом *SRI*.

1. Reconstruction/Demolition
   - выгода приравнивается к текущему риску: предполагается, что реконструкция/снос радикально снижает риск.
2. Strengthening
   - выгода определяется как снижение *SRI* до целевого уровня (*TARGET_SRI*), умноженное на вес грунтового риска: чем выше *SRI и GRI*, тем больше эффект от усиления.
3. Passportization
   - выгода отражает снижение неопределённости: считается, что паспортизация уменьшает информационный риск и позволяет корректно планировать дальнейшие мероприятия;
   - вводится коэффициент неопределённости (*UNCERTAINTY_WEIGHT*), который задаёт долю риска, «снимаемую» фактом обследования.

Для каждого здания оценивается базовая стоимость выбранной меры:

- для реконструкции/сноса: удельная стоимость на 1 м², умноженная на условную экспозицию (*exposure_m2*);
- для усиления: удельная стоимость усиления, скорректированная по уязвимости (V) и опасности (H), и умноженная на площадь;
- для паспортизации: фиксированная стоимость на объект.

В итоге каждое здание в рамках своей категории получает пару: *benefit* (сколько риска можно убрать) и *base_cost* (сколько это стоит).

Алматы қаласы инфрақұрылым басқармасы\АДәлдбай А. Ә.

```
# ===== Fast knobs =====
USE_DBSCAN = False          # set True when you can afford it
GRID_M = 350                # ~meters; buildings in same ~350m cell batch together
CRS_WGS84 = "EPSG:4326"
CRS_M = "EPSG:32643"        # UTM 43N ~ Almaty

# Costs (relative): keep simple & fast
COST_PER_M2_DEMO = 5.0
COST_PER_M2_STRENGTH = 3.0
COST_PER_PASSPORT = 0.05
TRAVEL_COST_PER_KM = 3000
TARGET_SRI = 0.30
UNCERTAINTY_WEIGHT = 0.8

def _clip01(x):
    return np.clip(pd.to_numeric(x, errors="coerce").fillna(0.0), 0.0, 1.0)

def ensure_geodf(df, geom_col="geometry", crs=CRS_WGS84):
    g = df.copy()
    if g[geom_col].dtype == object:
        mask = g[geom_col].notna()
        g.loc[mask, geom_col] = g.loc[mask, geom_col].apply(lambda v: wkt.loads(v) if isinstance
    g = gpd.GeoDataFrame(g, geometry=geom_col, crs=crs)
    bad = ~g.geometry.is_valid
    if bad.any():
        g.loc[bad, "geometry"] = g.loc[bad, "geometry"].buffer(0)
        g = g[g.geometry.notna() & g.geometry.is_valid]
    return g

def to_bool(s):
    if s.dtype == bool or str(s.dtype) == "bool":
        return s.fillna(False)
    txt = s.astype(str).str.lower().str.strip()
    t = {"true","1","yes","y","да","t"}; f = {"false","0","no","n","нет","none","nan","f"}
    out = np.where(txt.isin(t), True, np.where(txt.isin(f), False, np.nan))
    return pd.Series(out, index=s.index).fillna(False)

def exposure_m2(B):
    gfa = pd.to_numeric(B.get("gfa_m2", np.nan), errors="coerce")
    return np.where(gfa.notna() & (gfa>0), gfa,
                    B["area_m2"].fillna(0) * B["floor"].fillna(1).clip(lower=1))

def compute_action_scores(B, action):
    D = B.copy()
    D["SRI"] = _clip01(D["SRI"]); D["GRI"] = _clip01(D["GRI"])
    D["H"] = _clip01(D.get("H",0)); D["V"] = _clip01(D.get("V",0))
    D["exposure_m2"] = exposure_m2(D)
    D = D[D["exposure_m2"] > 0].copy()

    # SRI already derived from H, V, E in your pipeline.
    D["risk_now"] = D["SRI"] * (0.5 + 0.5*D["GRI"])

    if action == "Reconstruction/Demolition":
        D["benefit"] = D["risk_now"]
        D["base_cost"] = COST_PER_M2_DEMO * D["exposure_m2"]
        minutes = 90
    elif action == "Strengthening":
        D["benefit"] = np.maximum(0.0, D["SRI"] - TARGET_SRI) * (0.5 + 0.5*D["GRI"])
        D["base_cost"] = COST_PER_M2_STRENGTH * (1 + 0.5*D["V"] + 0.25*D["H"]) * D["exposure_m2"
        minutes = 60
    else:  # Passportization
        D["benefit"] = D["risk_now"] * UNCERTAINTY_WEIGHT
        D["base_cost"] = COST_PER_PASSPORT
        minutes = 30

    D["labor_cost"] = 0  # keep 0 for speed; plug if needed later
    D["minutes"] = minutes
    return D

def fast_grid_cluster(Gm, cell=GRID_M):
    # snap centroids to integer grid ids → cluster_id_grid
    c = Gm.geometry.centroid
    ix = np.floor(c.x.values / cell).astype(np.int64)
    iy = np.floor(c.y.values / cell).astype(np.int64)
```

Рисунок – 2.3.2.2. Код методологии

***Пространственная группировка и маршрутизация***

Для приоритизации не только «по риску», но и операционно-реалистичному показателю, введён блок пространственной группировки.

Здания переводятся в метрическую систему координат, после чего:

- либо разбиваются на ячейки грубой сетки фиксированного размера (например, ~350 м);
- либо, при необходимости, кластеризуются с помощью DBSCAN по

близости.

Результат — каждому зданию присваивается *cluster_id*, отражающий «батч» зданий, которые удобно обрабатывать совместно (одной бригаде, за один выезд).

Для каждого кластера рассчитывается:

- суммарная выгода (*benefit*) по всем зданиям;
- суммарная стоимость (*base_cost*);
- оценка «дорожной» составляющей — протяжённости маршрута по зданиям кластера;
- итоговая эффективность — отношение суммарной выгоды к полной оценочной стоимости (включая выезды).

Кластеры ранжируются по эффективности - сначала обрабатываются те группы зданий, в которых единица ресурса даёт максимальное снижение риска.

Каждому кластеру присваивается *cluster_rank*.

```python
def dbscan_cluster(Gm, eps_m, min_samples):
    cents = Gm.geometry.centroid
    XY = np.c_[cents.x.values, cents.y.values]
    labels = DBSCAN(eps=eps_m, min_samples=min_samples, n_jobs=-1).fit_predict(XY)
    return pd.Series(labels, index=Gm.index, name="cluster_id")

def route_km_quick(X):
    # fast proxy: cluster "diameter path" ~= perimeter/2 via bbox perimeter
    if len(X)==0: return 0.0
    minx, miny = X.min(0); maxx, maxy = X.max(0)
    perim = 2*((maxx-minx)+(maxy-miny))
    return (perim/2)/1000.0  # meters->km

def rank_batches_quick(Gm):
    out = []
    for cid, sub in Gm.groupby("cluster_id"):
        X = np.c_[sub.geometry.centroid.x.values, sub.geometry.centroid.y.values]
        km = route_km_quick(X)
        base = sub["base_cost"].sum()
        eff = base + TRAVEL_COST_PER_KM*km + sub["labor_cost"].sum()
        score = sub["benefit"].sum() / max(eff,1e-9)
        out.append((cid, score, km))
    # highest score first
    order = sorted(out, key=lambda t: t[1], reverse=True)
    cluster_rank = {cid:i+1 for i,(cid,_,_) in enumerate(order)}
    return pd.Series(Gm["cluster_id"].map(cluster_rank), index=Gm.index, name="cluster_rank")

def order_inside_cluster_quick(Gm):
    # sort by angle around cluster centroid (fast TSP-ish)
    out_rrk = np.empty(len(Gm), dtype=np.int32)
    for cid, sub in Gm.groupby("cluster_id"):
        idx = sub.index.values
        pts = np.c_[sub.geometry.centroid.x.values, sub.geometry.centroid.y.values]
        c = pts.mean(0)
        ang = np.arctan2(pts[:,1]-c[1], pts[:,0]-c[0])
        order = idx[np.argsort(ang)]
        # write 1..k back into the original rows for this cluster
        pos = {ix: i for i, ix in enumerate(order)}
        out_rrk[[pos[ix] for ix in sub.index.values]] = np.arange(1, len(order)+1, dtype=np.int32)
    return pd.Series(out_rrk, index=Gm.index, name="route_rank")

# ---------- NEW: flexible final sort ----------
def assign_priority(Gm, sort_by=None):
    """
    Assigns priority = 1..N after sorting by 'sort_by'.

    sort_by: list[tuple[str, 'asc'|'desc']]
        Examples:
          None/default: [("cluster_rank","asc"), ("route_rank","asc")]
          Risk-first:   [("cluster_rank","asc"), ("risk_now","desc"), ("SRI","desc"), ("route_rank","asc")]
          SRI-first:    [("cluster_rank","asc"), ("SRI","desc"), ("route_rank","asc")]
          Ignore clusters (set them to 1 beforehand) and do global risk: [("risk_now","desc"), ("SRI","desc")]

    Notes:
      - Missing columns are ignored silently.
      - Uses stable sort to keep previous order when keys tie.
    """
    if sort_by is None:
        sort_by = [("cluster_rank","asc"), ("route_rank","asc")]

    # keep only keys that exist
    by = []
    asc = []
    for col, direction in sort_by:
        if col in Gm.columns:
            by.append(col)
            asc.append(direction == "asc")

    if not by:  # fallback to index order if nothing valid is provided
        by, asc = [Gm.index.name or "index"], [True]
        Gm = Gm.reset_index().rename(columns={"index":"__idx__"})
        by, asc = ["__idx__"], [True]

    Gm = Gm.sort_values(by=by, ascending=asc, kind="mergesort")
```

Рисунок – 2.3.2.3. Пространственная группировка и маршрутизация

Внутри каждого кластера здания упорядочиваются в приблизительный маршрут (аналог упрощённого решения TSP):

- рассчитывается центроид кластера;

– здания упорядочиваются по углу относительно центра (по полярному углу), что даёт компактный маршрут обхода.

Каждое здание внутри кластера получает *route_rank* — порядок посещения.

***Назначение глобального приоритета***

Финальный шаг — присвоение каждому зданию глобального приоритета в пределах своей категории мероприятия.

Порядок задаётся набором ключей сортировки, например:

– сначала *cluster_rank* (какие кластеры важнее),
– затем показатели риска (*risk_now, SRI*),
– затем *route_rank* (удобный порядок обхода внутри кластера).

В результате каждому зданию присваивается числовой приоритет (priority = 1, 2, 3, …), который:

– учитывает как риск, так и стоимость и логистику;
– позволяет формировать упорядоченные списки для средне- и долгосрочных планов.

```python
def process_category(B, action, eps_m_dbscan, sort_by=None):
    cols = ["building_id","priority","cluster_id","cluster_rank","route_rank",
            "benefit","base_cost","risk_now","exposure_m2","SRI","GRI"]
    if len(B)==0:
        return pd.DataFrame(columns=cols)

    S = compute_action_scores(B, action)
    Gm = S.to_crs(CRS_M)

    # ultra-fast spatial batching
    if USE_DBSCAN:
        # use coarser eps for speed per action; min_samples small
        min_samples = {"Passportization":4, "Strengthening":3, "Reconstruction/Demolition":2}[action]
        Gm["cluster_id"] = dbscan_cluster(Gm, eps_m=eps_m_dbscan, min_samples=min_samples)
        # treat noise as singleton clusters
        noise = Gm["cluster_id"] == -1
        if noise.any():
            Gm.loc[noise, "cluster_id"] = (Gm.loc[noise].index.values.astype(np.int64) + 10 + 7)
    else:
        Gm["cluster_id"] = fast_grid_cluster(Gm, cell=GRID_M)

    # rank clusters quickly
    Gm["cluster_rank"] = rank_batches_quick(Gm)

    # order inside cluster (proxy route)
    Gm["route_rank"] = order_inside_cluster_quick(Gm)

    # ---- global per-category priority with flexible sort ----
    # Examples for sort_by:
    #   None -> default (cluster_rank asc, route_rank asc)
    #   [("cluster_rank","asc"), ("risk_now","desc"), ("SRI","desc"), ("route_rank","asc")]
    #   [("cluster_rank","asc"), ("SRI","desc"), ("route_rank","asc")]
    Gm = assign_priority(Gm, sort_by=sort_by)

    return Gm[cols]

# ================= MAIN =================
def build_fast_priorities(df, sort_by=None):
    """
    build priorities per category with flexible final ordering.

    sort_by:
      - list of (column, 'asc'|'desc') applied to each category:
          e.g. [("cluster_rank","asc"), ("risk_now","desc"), ("SRI","desc"), ("route_rank","asc")]
      - None -> default [("cluster_rank","asc"), ("route_rank","asc")]
    """
    B = ensure_geodf(df, "geometry", CRS_WGS84)
    B["is_emergency_building"] = to_bool(B["is_emergency_building"])
    B["is_passport"] = to_bool(B["is_passport"])
    B["seismic_eval"] = to_bool(B["seismic_eval"])

    cond_demo = B["is_emergency_building"]
    cond_pass = ~B["is_passport"]
    cond_str  = B["is_passport"] & (~B["seismic_eval"])

    # measure_category
    B["measure_category"] = np.select(
        [cond_demo, cond_pass, cond_str],
        ["Reconstruction/Demolition","Passportization","Strengthening"],
        default="stable case"
    )

    parts = []
    parts.append(process_category(B[cond_demo], "Reconstruction/Demolition", eps_m_dbscan=700, sort_by=sort_by))
    parts.append(process_category(B[cond_pass], "Passportization",           eps_m_dbscan=350, sort_by=sort_by))
    parts.append(process_category(B[cond_str],  "Strengthening",             eps_m_dbscan=500, sort_by=sort_by))

    P = pd.concat(parts, ignore_index=False)
    out = B.merge(P, on="building_id", how="left")

    # return same schema as input + new columns
    return gpd.GeoDataFrame(out, geometry="geometry", crs=B.crs)
```

Рисунок – 2.3.2.4. Назначение глобального приоритета

***Визуализация приоритетов реконструкции/сноса***

Для категории *Reconstruction/Demolition* дополнительно построена интерактивная карта:

– используются центроиды зданий в системе координат WGS84;

Алматы қаласы ... АДандібай А. Э.

- каждому объекту присваивается цвет маркера по шкале от зелёного (низкий приоритет) до красного (высокий приоритет);
- размер маркера варьируется в зависимости от приоритета, чтобы наиболее важные объекты были визуально выделены;
- при клике по объекту отображается всплывающее окно с ключевой информацией:
    - приоритет,
    - идентификатор здания,
    - значение SRI,
    - район.

Такая карта может быть интегрирована в аналитическую панель ситуационного центра и использоваться для пошагового планирования программы реконструкции и сноса.

***Выводы по модулю сейсмических рекомендаций***

Построен единый алгоритм, который переводит абстрактные индексы риска (SRI, GRI) в конкретные приоритетные списки объектов по типам мероприятий.

В расчёте учтены не только значения риска, но и стоимость вмешательства и логистика выездов, что делает приоритизацию практически применимой.

Предусмотрено разделение на три ключевые категории: паспортизация, усиление, реконструкция/снос, плюс стабильные объекты.

Разработан интерактивный картографический слой для визуализации приоритетов, что облегчает коммуникацию с городскими службами и принятие решений.

***Макет аналитического модуля «Мониторинг сейсмоустойчивости городских объектов»***

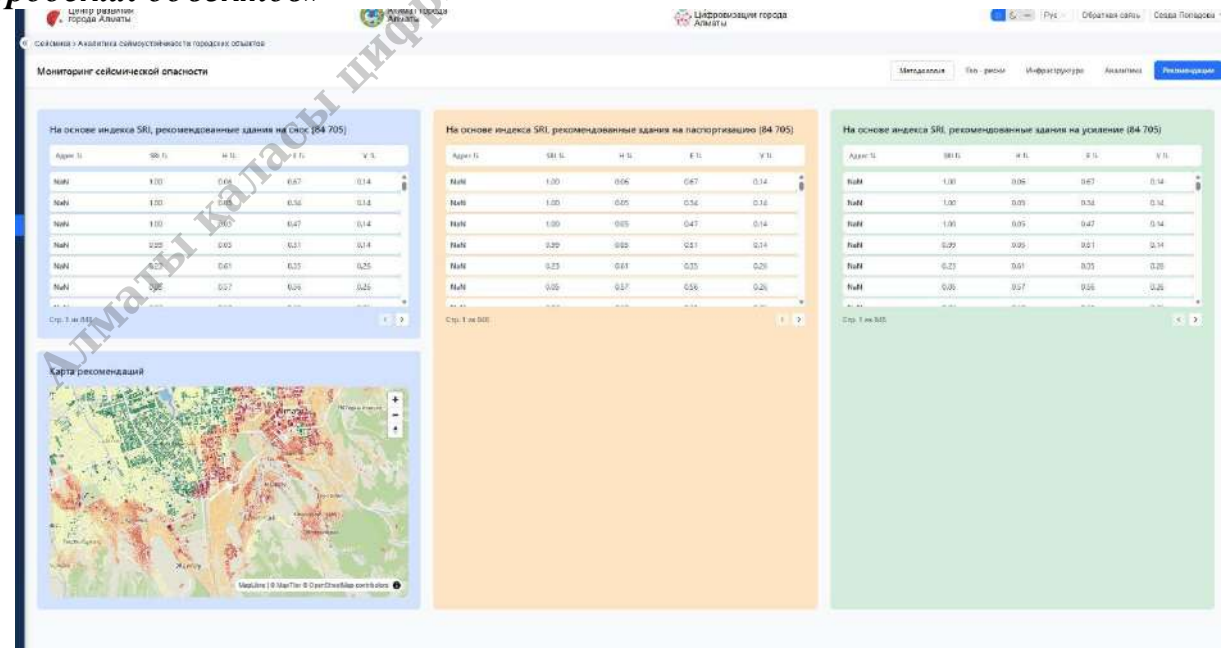



Рисунок – 2.3.2.5. Макет аналитического модуля «Мониторинг сейсмоустойчивости городских объектов»

Алматы қаласы цифрландыру басқармасы\Аданібай А.Ә.

## 2.5.Разработка модуля «Мониторинг сейсмической опасности»

***Цель:***

Обеспечить мониторинг оползнеопасных участков, сейсмоактивных зон и оперативных данных о сейсмических событиях для повышения готовности к чрезвычайным ситуациям.

***Задачи:***

4-й квартал:

- Настройка системы формирования ежедневной отчетности о сейсмических событиях.

***Источник данных:*** Национальный научный центр сейсмологических наблюдений и исследований, Управление сейсмики города Алматы.

***Заказчик:*** Управление сейсмики города Алматы, Национальный научный центр сейсмологических наблюдений и исследований.

Ранее была разработана аналитический модуль «Мониторинг сейсмической опасности», состоящая из следующих страниц:

- Гео-риски;
- Инфраструктура;
- Аналитика.

Дашборд отображает координаты геоструктур (оползни, тектонические разломы, селевые пути), уровень георисков, плотность населения, готовность инфраструктуры и сейсмостойкость зданий. Реализованы фильтры по районам, уровням риска, категориям геоструктур, инженерным узлам и социальным объектам.

В текущем квартале были проведены работы по улучшению модуля:

На каждой из страниц, были добавлены методологии с подробным описанием логики расчетов и применяемых формул при формировании показателей (Рисунок – 2.3.3.1 – 2.3.3.3).

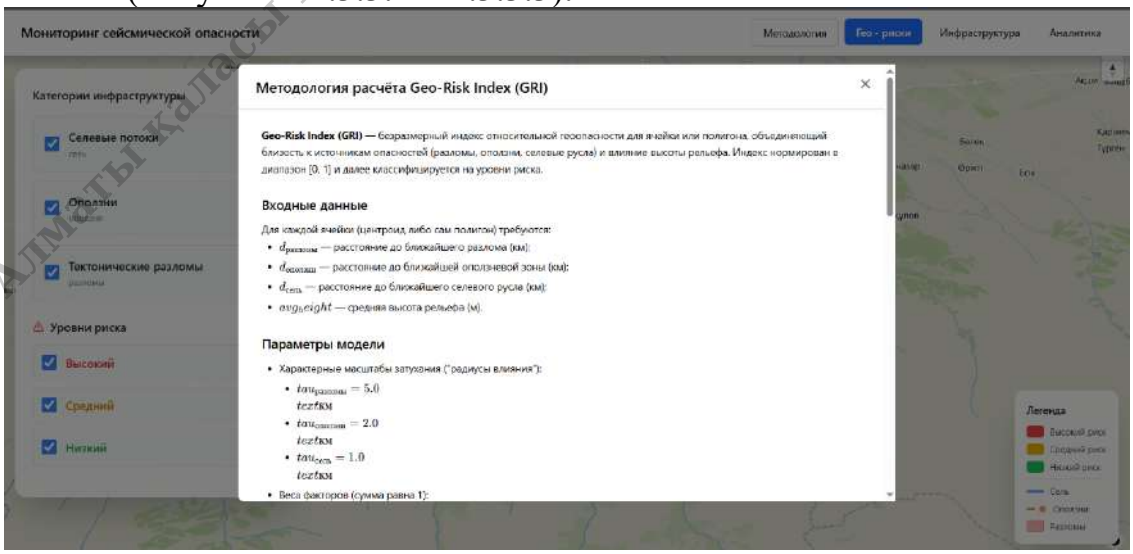



Рисунок – 2.3.3.1. Методология расчета «Гео-риски»

Алматы қаласы инфрландыру басқармасы\Ад әндібай А. Ә.

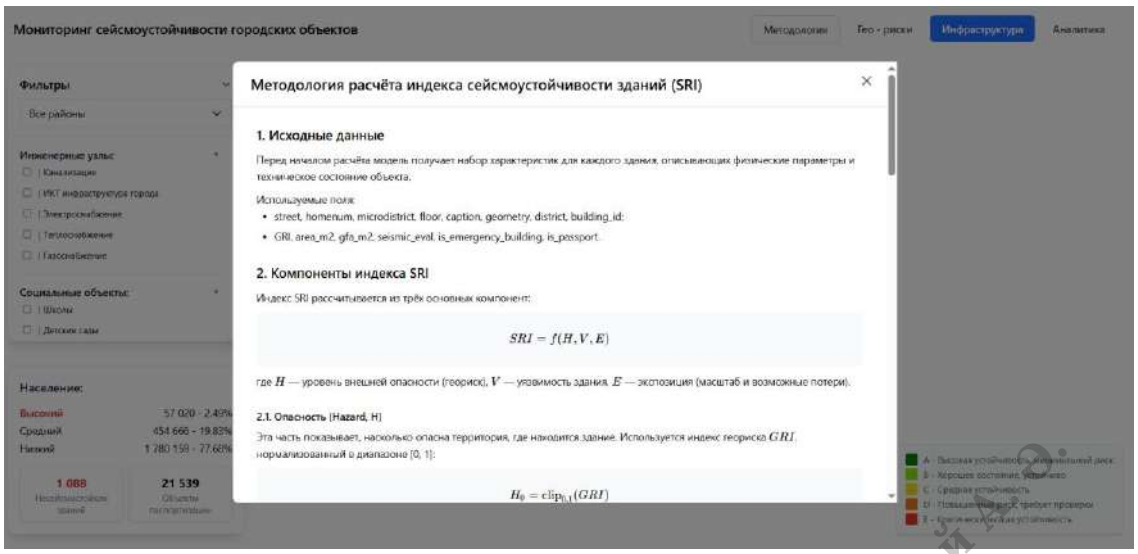
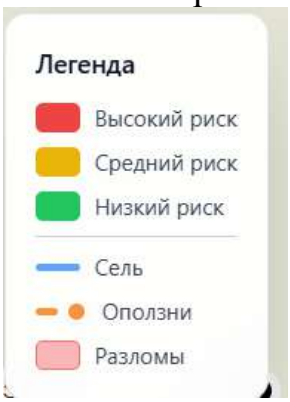


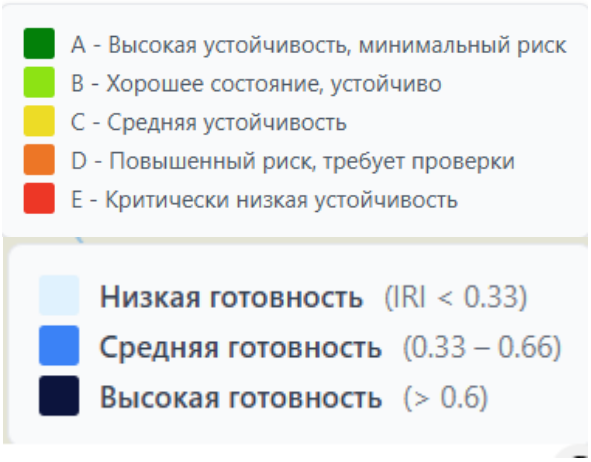

Рисунок – 2.3.3.2. Методология расчета «Инфраструктура» (Риск зданий)

Рисунок – 2.3.3.3. Методология расчета «Инфраструктура» (Готовность инфраструктуры)

Добавлены **легенды для карт**, поясняющие цветовые обозначения, категории объектов и диапазоны значений георисков.




Алматы қаласы цифрландыру басқармасы АДәндібай Ә.Ә.




Проведена общая оптимизация отображения данных для повышения наглядности и удобства анализа.

### 2.5.1. Настройка системы формирования ежедневной отчетности о сейсмических событиях

В рамках расширения функционала мониторинга сейсмической активности города была внедрена автоматизированная система формирования и рассылки ежедневной отчетности о зарегистрированных сейсмических событиях. Система использует данные, поступающие из модуля «Мониторинг сейсмоустойчивости городских объектов» и интегрированных внешних источников сейсмологических наблюдений.

На основе поступающих данных реализован процесс регулярной выборки событий за последние сутки с учетом ключевых параметров: дата и время землетрясения, магнитуда, глубина, координаты эпицентра, расстояние до города, классификация по уровню опасности, а также наличие потенциального воздействия на городские объекты. Для обеспечения корректности формирования отчетов внедрен механизм валидации данных, включающий проверку полноты, формата и отсутствия дублирующих записей.

Система автоматически формирует ежедневный сводный отчет в формате Excel, включающий:

- перечень всех сейсмических событий за прошедшие 24 часа;
- их ключевые характеристики;
- аналитические показатели (классификация по уровню угрозы, распределение по времени суток, динамика по сравнению с предыдущим периодом);
- визуальные элементы (условные цветовые метки уровней опасности, диаграммы при необходимости);
- список объектов городской инфраструктуры, попадающих в зону возможного воздействия согласно георадиусным моделям.

Генерация Excel-файла осуществляется на базе заранее подготовленного шаблона с автоматическим заполнением структурированных таблиц и параметризованных полей. Далее отчет автоматически передается в почтовый

Алматы халасы цифрлендыру басқармасы\АДәндібай А. Ә.

модуль, где с использованием SMTP-шлюза выполняется рассылка на утвержденный перечень адресатов Управления городского планирования, ЧС и других профильных подразделений.

В рамках настройки системы выполнены следующие технические работы:

- разработка SQL/ETL-процедаур для выборки и очистки данных;
- автоматизация ежедневного запуска через систему планировщиков (cron/Task Scheduler/Airflow);
- интеграция с модулем формирования Excel-документов;
- настройка почтового клиента и параметров SMTP;
- логирование всех этапов (выборка, валидация, генерация отчетов, отправка писем);
- внедрение механизмов отслеживания ошибок и повторных попыток при недоставке писем.

Система позволяет ответственным структурным подразделениям получать актуальные сведения о сейсмической активности в строгом ежедневном режиме, обеспечивая оперативность принятия решений и повышение готовности к реагированию.

### 2.6.Разработка модуля «Мониторинг договоров»

**Цель**: Разработка инструмента для мониторинга строительно-ремонтных договоров.

**Задачи 4-квартала:**

- Добавление в форму регистрации данных функционала по регистрации договоров на разработку ПСД**.**
- Реализация мониторинга и анализа данных по договорам на строительство с выявлением аффилированных компаний (совпадение реквизитов, пользователей, счетов, совместные участия)
- Аналитика по заказчикам (закупки без конкуренции, ротации победителей, РНП-риски, временные аномалии).
- Реализация мониторинга и анализа данных по договорам ПСД

**Источники данных**: Портал государственных закупок

**Заказчик**: Департамент экономических расследований.

#### 2.6.1. Добавление в форму регистрации данных функционала по регистрации договоров на разработку ПСД

В 4-квартале были проведены работы по оптимизации работ парсера при сборе данных с портала Государственных закупок, при помощи внедрения новой архитектуры с использованием GraphQL, что позволило повысить скорость загрузки данных.

Алматы қаласы цифрландыру басқармасы\АЛәндібай А. Ә.

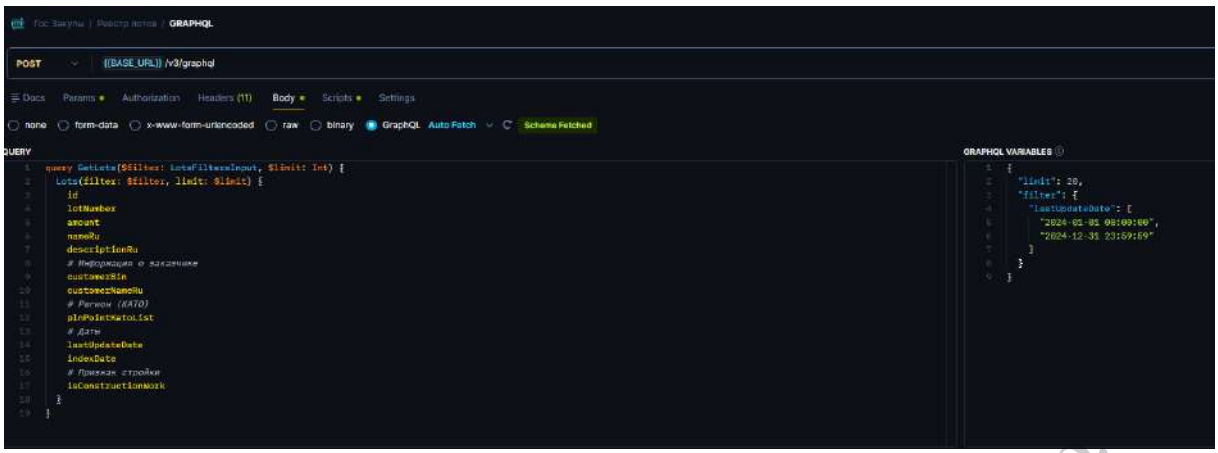



Рисунок – 2.3.4.1. Запрос GraphQL в Postman

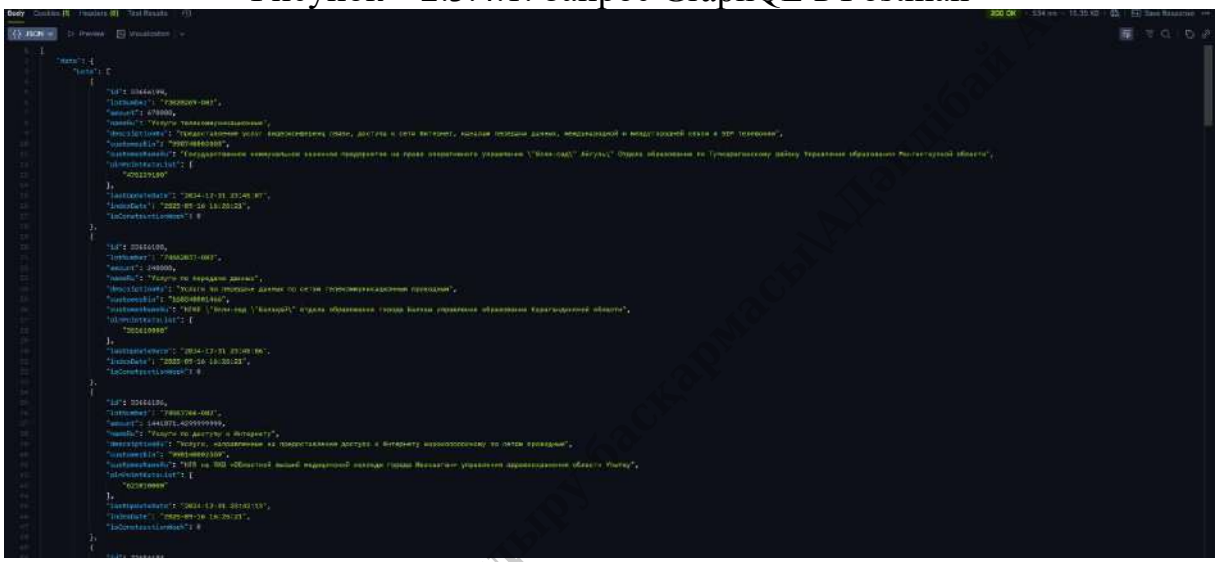
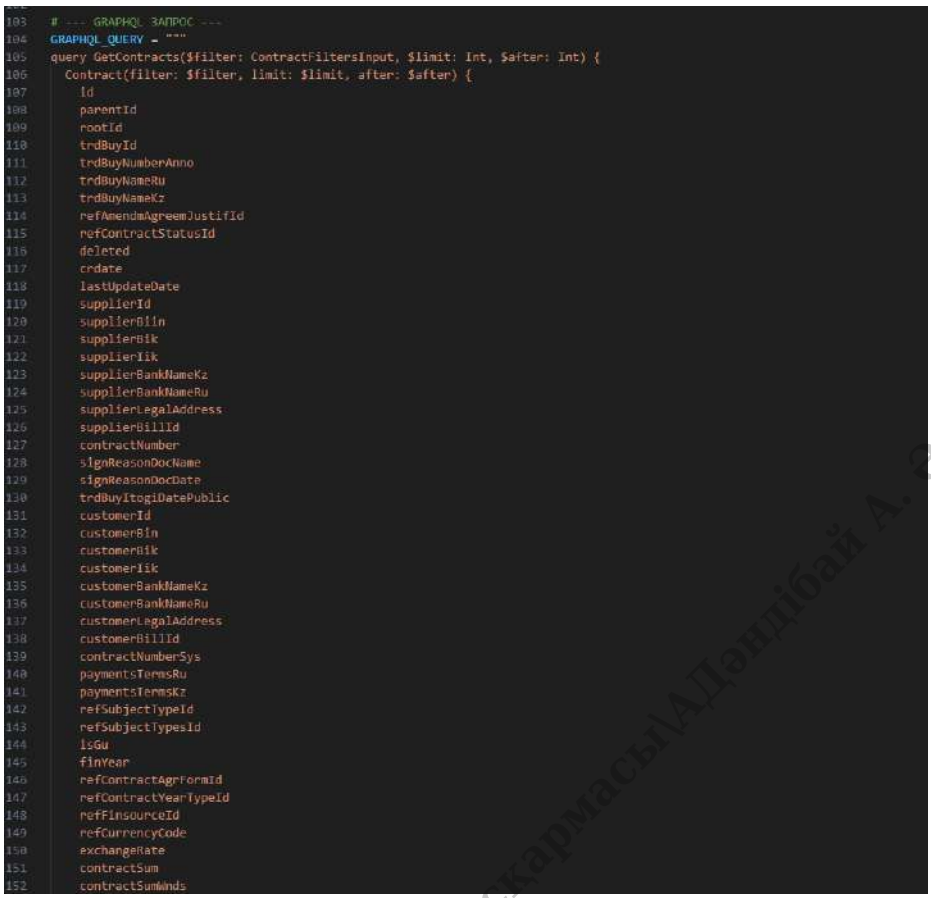


Рисунок – 2.3.4.2. Ответ GraphQL в Postman

Также были проведены работы по оптимизации функции фильтрации данных, что позволит запрашивать у источника данных договоры, связанные со строительством и ремонтом.

Алматы қаласы киберлаңы



```
# --- GRAPHQL ЗАПРОС ---
GRAPHQL_QUERY = """
query GetContracts($filter: ContractFiltersInput, $limit: Int, $after: Int) {
  Contract(filter: $filter, limit: $limit, after: $after) {
    id
    parentId
    rootId
    trdBuyId
    trdBuyNumberAnno
    trdBuyNameRu
    trdBuyNameKz
    refAmendmAgreemJustifId
    refContractStatusId
    deleted
    crdate
    lastUpdateDate
    supplierId
    supplierBiin
    supplierBik
    supplierIik
    supplierBankNameKz
    supplierBankNameRu
    supplierLegalAddress
    supplierBillId
    contractNumber
    signReasonDocName
    signReasonDocDate
    trdBuyItogiDatePublic
    customerId
    customerBin
    customerBik
    customerIik
    customerBankNameKz
    customerBankNameRu
    customerLegalAddress
    customerBillId
    contractNumberSys
    paymentsTermsRu
    paymentsTermsKz
    refSubjectTypeId
    refSubjectTypesId
    isGu
    finYear
    refContractAgrFormId
    refContractYearTypeId
    refFinsourceId
    refCurrencyCode
    exchangeRate
    contractSum
    contractSumWnds
```

Рисунок – 2.3.4.3. Фильтрация данных в скрипте

Для ускорения процесса обработки больших объемов данных, были проведены работы по внедрению технологии контейнеризация. Для внедрения данной технологии с помощью docker-compose были развернуты три параллельных сервиса, каждый из которых обрабатывает отдельный временной период (2023, 2024, 2025 годы). Данная оптимизация позволило сократить время полного обновления базы данных с нескольких суток до нескольких часов.

Алматы қаласы цифрлендыру басқармасы\АДәндібай А. Ә.

Рисунок – 2.3.4.4. Контейнеризация в docker-compose

### 2.6.2. Реализация мониторинга и анализа данных по договорам на строительство с выявлением аффилированных компаний

Цель выполнения данной задачи - обеспечить прозрачность и контроль в строительных закупках путем автоматического выявления аффилированных компаний, которые могут участвовать в одной закупке под разными юридическими лицами.

Это позволяет обнаруживать риски искусственной конкуренции, сговора, завышения стоимости и возможные связи с компаниями из РНП, повышая качество мониторинга и предотвращая коррупционные схемы.

Для решения этой задачи был создан алгоритм, позволяющий выявлять компании, участвующие в одной закупке под разными юридическими лицами. Он анализирует все заявки внутри одного объявления (buy_id) и определяет возможную аффилированность на основе набора признаков: совпадение номеров счетов, ФИО подающих заявку, корня фамилии, e-mail, телефонных номеров и веб-сайта. Перед сравнением все контактные данные проходят нормализацию.

Ақпарат қауіпсіздігі департаментінің басқармасы\Адаңдібай А. Ә.

Помимо этого, проводится проверка на связи с Реестром недобросовестных поставщиков (РНП) — как по прямому совпадению, так и по совпадающим контактам. На основе объединённых данных алгоритм выявляет компании, которые могут быть связаны друг с другом или с организациями из РНП, и формирует итоговый перечень подозрительных участников строительных и ремонтных закупок.

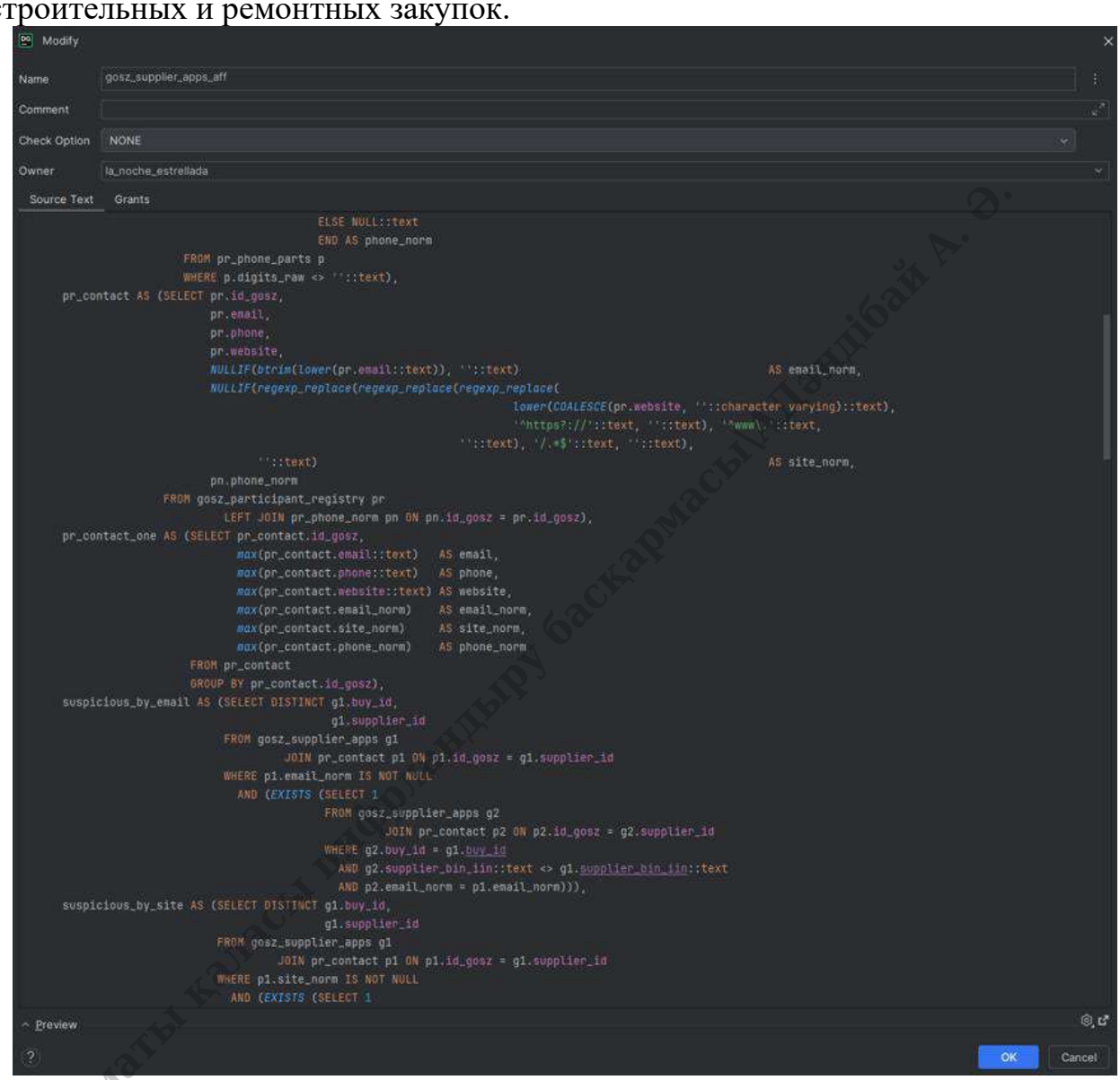



Рисунок – 2.3.4.5. Фрагмент расчета Аффилированности

Алматы қаласы тұрғындыру басқармасы Таңдібай А. Ә.

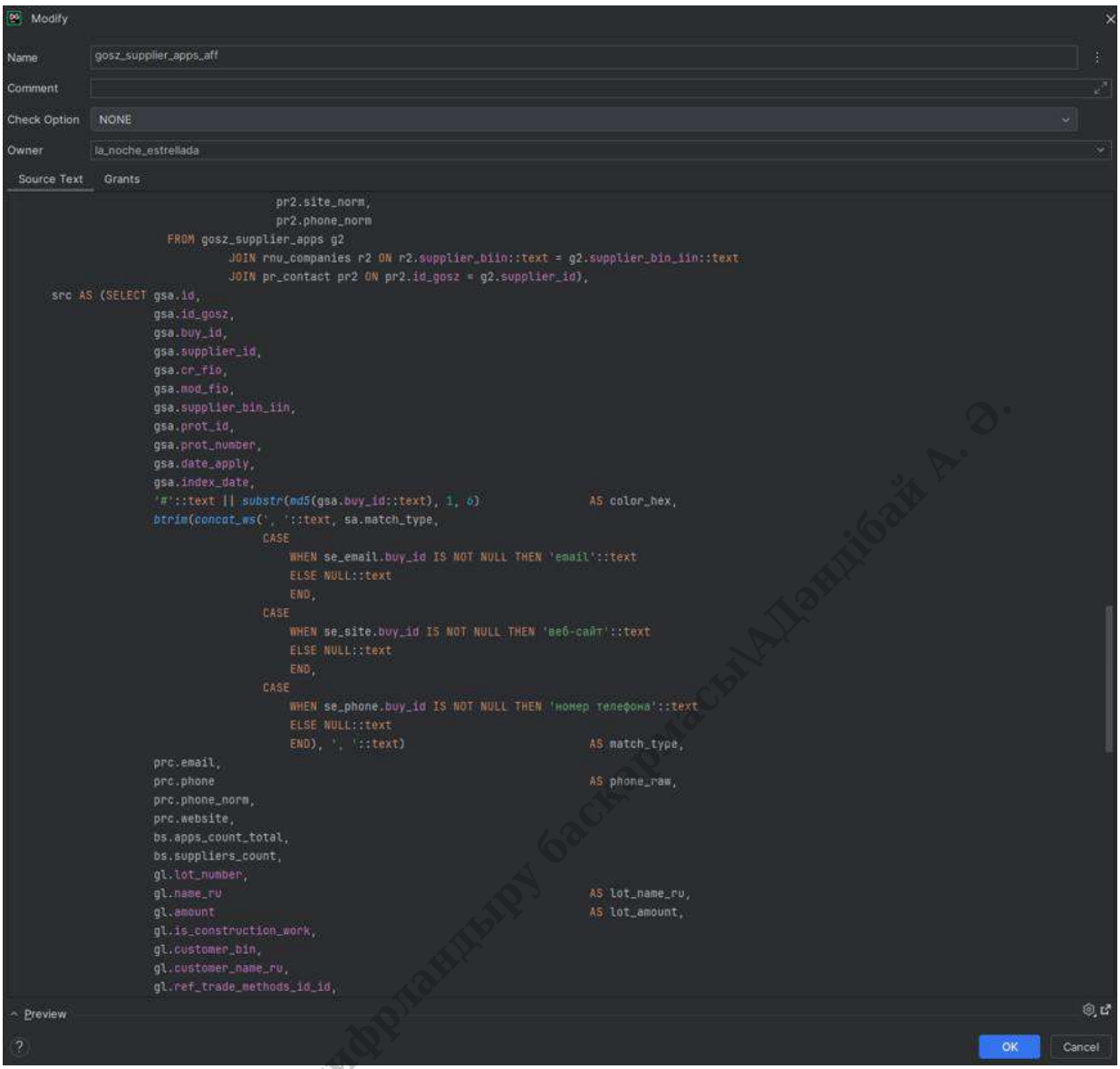



Рисунок – 2.3.4.6. Фрагмент расчета Аффилированности

Дополнительно в анализ включён модуль совместных участий, который позволяет обнаруживать пары компаний, регулярно подающих заявки на одни и те же закупочные объявления. Для каждой такой пары рассчитывается количество совместных заявок, частота побед каждой из компаний и доля выигранных закупок в случаях их одновременного участия.

Повторяющиеся совместные участия, особенно при заметно неравномерном распределении побед, могут свидетельствовать о фиктивной конкуренции, согласованных действиях или скрытой аффилированности между компаниями. Подобные связи увеличивают общий риск-профиль участника, особенно если одна из сторон уже включена в РНП или предварительно отмечена как подозрительная по контактным признакам.

Анализ совместных участий позволяет всесторонне оценивать возможные схемы координированных действий, выявлять устойчивые пары «конкурентов», которые систематически участвуют вместе в строительных закупках, и повышать прозрачность процесса за счёт раннего обнаружения потенциальных

Алматы қаласы цифрландыру басқармасы\АДәндібай А. Ә.

картельных практик.

Modify

Name: gosz_joint_participation_comp

Comment:

Check Option: NONE

Owner: la_noche_estrellada

Source Text | Grants

```
WITH apps AS (SELECT DISTINCT gosz_supplier_apps.buy_id,
                              gosz_supplier_apps.supplier_bin_iin
              FROM gosz_supplier_apps),
     pairs AS (SELECT a1.buy_id,
                      LEAST(a1.supplier_bin_iin, a2.supplier_bin_iin)    AS sup_a,
                      GREATEST(a1.supplier_bin_iin, a2.supplier_bin_iin) AS sup_b
               FROM apps a1
                        JOIN apps a2
                             ON a1.buy_id = a2.buy_id AND a1.supplier_bin_iin::text < a2.supplier_bin_iin::text),
     lot_winner AS (SELECT l.trd_buy_id    AS buy_id,
                           c.supplier_biin AS winner_bin
                    FROM gosz_lots l
                             JOIN gosz_lot_status ls ON ls.id_gosz = l.ref_lot_status_id_id
                             LEFT JOIN gosz_contract_units cu ON cu.lot_id = l.id_gosz
                             LEFT JOIN gosz_contract c ON c.id_gosz = cu.contract_registry_id
                    WHERE ls.name_ru::text = 'Закупка состоялась'::text),
     base AS (SELECT DISTINCT p.sup_a,
                              p.sup_b,
                              p.buy_id,
                              w.winner_bin
              FROM pairs p
                       LEFT JOIN lot_winner w ON w.buy_id = p.buy_id),
     names AS (SELECT COALESCE(gosz_participant_registry.bin, gosz_participant_registry.iin) AS bin,
                      gosz_participant_registry.name_ru
               FROM gosz_participant_registry)
SELECT b.sup_a,
       na.name_ru                                                                    AS sup_a_name,
       b.sup_b,
       nb.name_ru                                                                    AS sup_b_name,
       (b.sup_a::text || '|'::text) || b.sup_b::text                                 AS pair_key,
       count(DISTINCT b.buy_id)                                                      AS joint_lots,
       count(DISTINCT b.buy_id) FILTER (WHERE b.winner_bin::text = b.sup_a::text)    AS wins_by_a,
       count(DISTINCT b.buy_id) FILTER (WHERE b.winner_bin::text = b.sup_b::text)    AS wins_by_b,
       count(DISTINCT b.buy_id)
       FILTER (WHERE b.winner_bin::text = b.sup_a::text OR b.winner_bin::text = b.sup_b::text) AS wins_by_pair,
       round(count(DISTINCT b.buy_id)
             FILTER (WHERE b.winner_bin::text = b.sup_a::text OR b.winner_bin::text = b.sup_b::text)::numeric * 100.0 /
             NULLIF(count(DISTINCT b.buy_id), 0)::numeric, 1)                        AS pair_win_share_pct
FROM base b
         LEFT JOIN names na ON na.bin::text = b.sup_a::text
         LEFT JOIN names nb ON nb.bin::text = b.sup_b::text
GROUP BY b.sup_a, na.name_ru, b.sup_b, nb.name_ru
HAVING count(DISTINCT b.buy_id) >= 5
```

Preview

Рисунок – 2.3.4.7. Фрагмент расчета совместных участий

Алматы қаласы цифрландыру басқармасы\АДәндібай А. Ә.

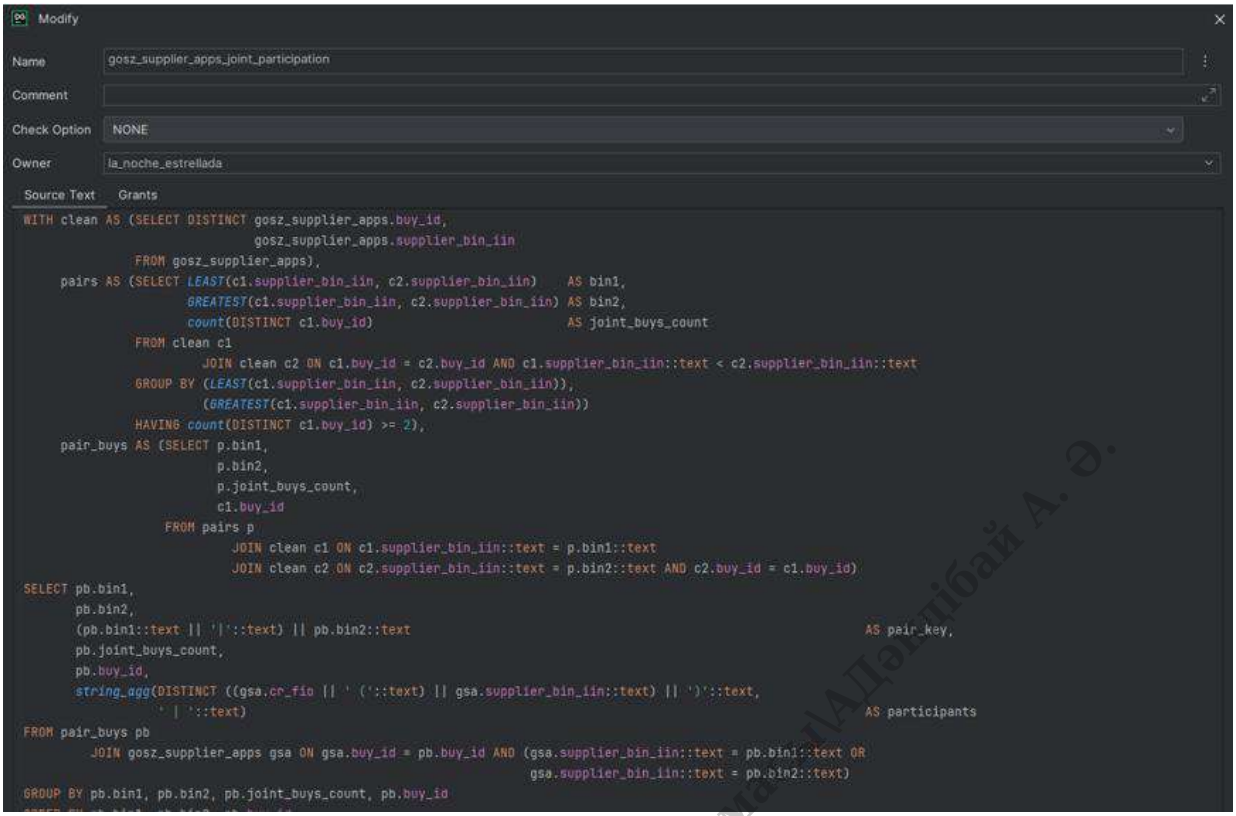



Рисунок – 2.3.4.8. Фрагмент расчета совместных участий

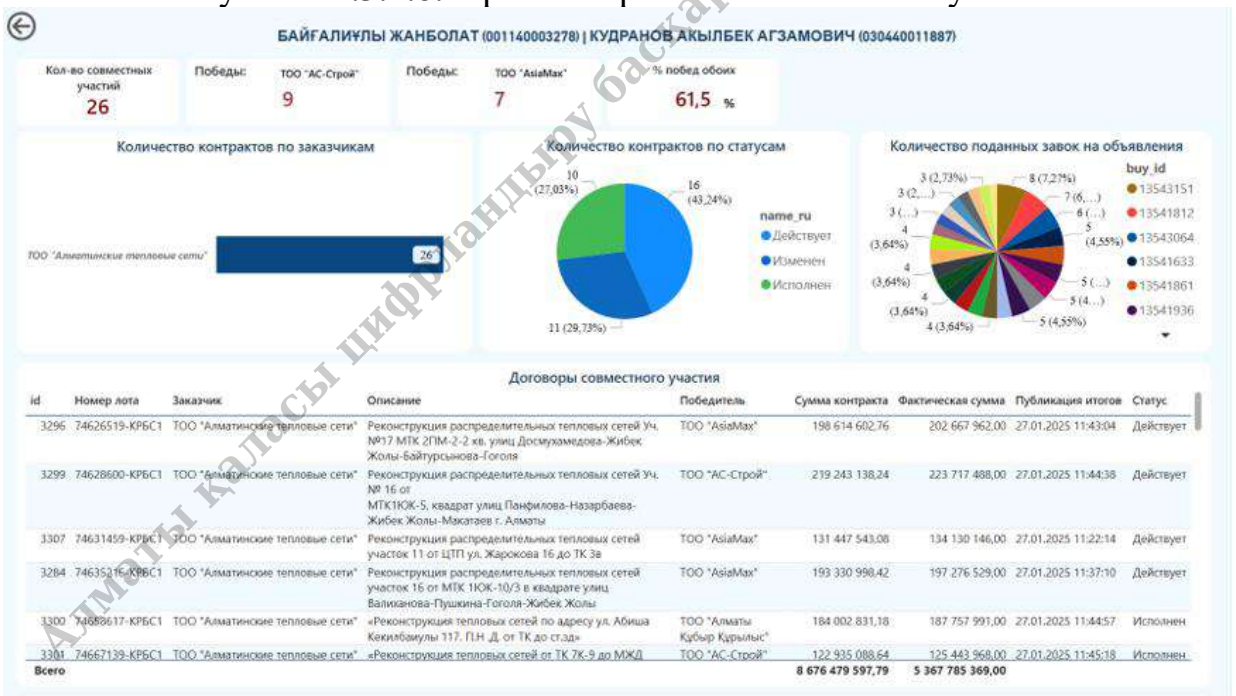



Рисунок – 2.3.4.9. Дашборд «Детализация совместных участий»

### 2.6.3. Аналитика по заказчикам

Для повышения прозрачности закупочных процедур и управления рисками создан аналитический модуль, который позволяет оценивать работу заказчиков по ряду ключевых показателей: доля неконкурентных закупок, ротация победителей, взаимодействие с поставщиками из РНП, а также выявление

Алматы қаласы цифрландыру басқармасы\Әдәлібай А. Ә.

временных аномалий в процессе проведения процедур.

Модуль предоставляет целостное представление о закупочной деятельности заказчиков и служит основой для принятия обоснованных управленческих решений.

***Закуп без конкуренции***

Система выявляет объявления, по которым была подана только одна заявка либо процедура проведена в упрощённом формате.

Для каждого заказчика отображаются:

- количество неконкурентных закупок;
- их доля в общем числе процедур;
- перечень договоров с указанием победителя, суммы, даты подачи заявки и даты публикации результатов.

Такой анализ помогает обнаруживать заказчиков с высокой долей закупок без конкуренции, что может свидетельствовать о низкой рыночной активности, ограниченном пуле поставщиков или возможных договорённостях.

***Ротации победителей***

Проводится анализ повторяемости побед одних и тех же поставщиков у каждого конкретного заказчика.

Модуль показывает:

- количество заключённых контрактов по каждому подрядчику;
- частоту повторяющихся побед;
- список закупок с указанием числа заявок и сумм заключённых договоров.

Отсутствие ротации победителей может указывать на снижение уровня конкуренции, устойчивые предпочтения заказчика или возможную аффилированность между заказчиком и поставщиками.

***РНП – риски***

Система выявляет случаи заключения договоров заказчиками с поставщиками:

- включенными в Реестр недобросовестных поставщиков (прямое попадание);
- связанными с компаниями из РНП через контактные данные, учредителей или совместные участия.

Для каждого заказчика отображается количество контрактов с повышенными РНП-рисками.

Такой анализ позволяет оценить качество отбора контрагентов и соблюдение требований добросовестности.

***Временные аномалии***

В рамках временного анализа система сравнивает ключевые даты:

- дату подачи заявки;
- дату публикации итогов;
- дату заключения договора.

Алматы қаласы цифрландыру басқармасы АДэлібай А. Ә.

Обнаруживаются аномально короткие или длительные промежутки между этапами закупки, которые могут свидетельствовать о:

- ускоренные процедуры;
- неполную конкуренцию;
- согласованное поведение участников;
- организационные нарушения в процессе проведения торгов.

Modify

Name: gosz_winners

Comment:

Check Option: NONE

Owner: la_noche_estrellada

Source Text | Grants

```
WITH winner AS (SELECT l.id_gosz          AS lot_id,
                       l.trd_buy_id       AS buy_id,
                       l.customer_id,
                       l.customer_bin,
                       l.customer_name_ru,
                       l.last_update_date AS lot_last_update_date,
                       c.contract_registry_id,
                       ct.supplier_biin   AS winner_bin,
                       ct.supplier_id     AS winner_id,
                       pr.full_name_ru    AS supplier_full_name,
                       pr.name_ru         AS supplier_name_ru,
                       ct.contract_sum    AS winner_sum
                FROM gosz_lots l
                         LEFT JOIN gosz_contract_units c ON c.lot_id = l.id_gosz
                         LEFT JOIN gosz_contract ct ON ct.id_gosz = c.contract_registry_id
                         LEFT JOIN gosz_participant_registry pr ON pr.id_gosz = ct.supplier_id
                         JOIN gosz_lot_status ls ON ls.id_gosz = l.ref_lot_status_id_id
                WHERE ls.name_ru::text = 'Закупка состоялась'::text),
     bidders AS (SELECT gosz_supplier_apps.buy_id,
                        count(DISTINCT gosz_supplier_apps.supplier_bin_iin) AS bidders_cnt
                 FROM gosz_supplier_apps
                 GROUP BY gosz_supplier_apps.buy_id),
     rnu AS (SELECT DISTINCT gosz_rnu.supplier_biin AS rnu_bin
             FROM gosz_rnu)
SELECT w.lot_id,
       w.buy_id,
       w.customer_id,
       w.customer_bin,
       w.customer_name_ru,
       w.lot_last_update_date,
       w.contract_registry_id,
       w.winner_bin,
       w.winner_id,
       w.supplier_full_name,
       w.supplier_name_ru,
       w.winner_sum,
       b.bidders_cnt,
       CASE
           WHEN r.rnu_bin IS NOT NULL THEN 1
           ELSE 0
           END AS winner_in_rnu
FROM winner w
```

Рисунок – 2.3.4.10. Фрагмент кода с расчетом победителей, поданных заявок

Алматы қаласы цифрландыру басқармасы\Адәндібай А. Ә.

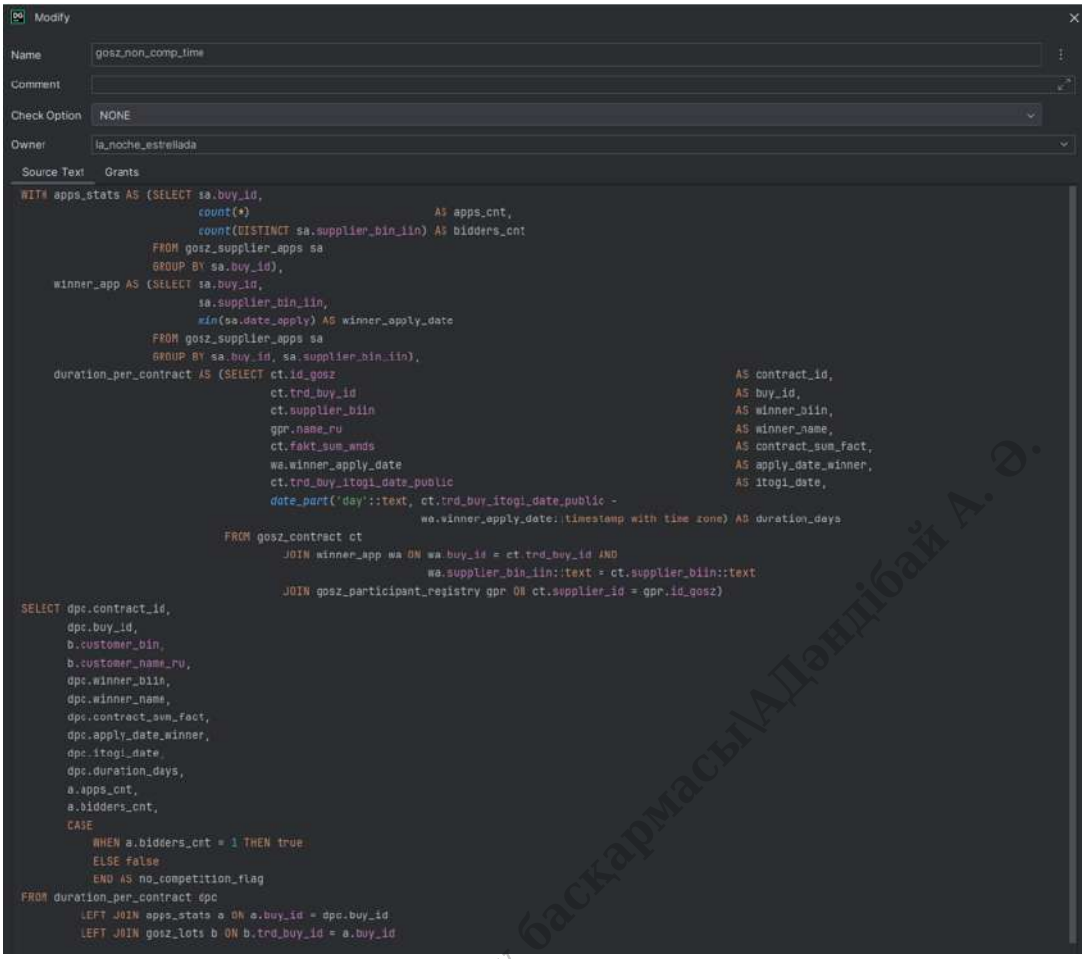



Рисунок – 2.3.4.11. Фрагмент кода с расчетом закупок без конкуренций

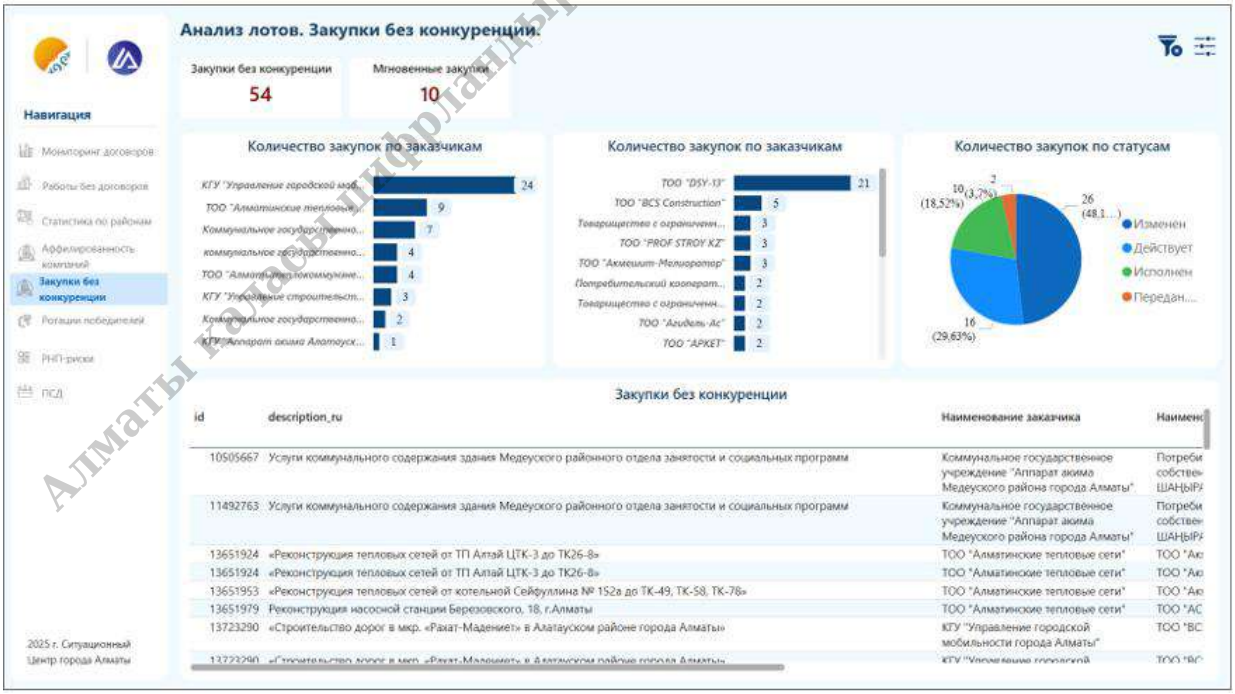



Рисунок – 2.3.4.12. Дашборд с закупками без конкуренции, с временными аномалиями

Алматы қаласы цифрландыру басқармасы\Адәндібай А. Ә.

Рисунок – 2.3.4.13. Дашборд с ротацией победителей, РНП-рисками

### 2.6.4. Реализация мониторинга и анализа данных по договорам ПСД

В рамках мониторинга проектно-сметной документации (ПСД) разработан аналитический модуль, обеспечивающий комплексный контроль по всем заключённым договорам ПСД в разрезе заказчиков, статусов выполнения и динамики по годам.

Система автоматически собирает и агрегирует данные из закупочной системы, формируя единое окно для анализа объёмов ПСД, текущего состояния договоров и ключевых тенденций.

Модуль позволяет:

**1. Оценивать общий объём договоров ПСД**

Отображается общее количество заключённых договоров и количество действующих ПСД, что обеспечивает контроль текущей загрузки и своевременности выполнения работ.

**2. Анализировать ПСД по заказчикам**

Дашборд показывает крупнейших заказчиков ПСД, объёмы заключённых договоров и распределение работ между организациями. Это позволяет выявлять центры концентрации расходов и планировать нагрузку.

**3. Контролировать статусы выполнения договоров**

Отображается распределение договоров по статусам: действует, исполнен, частично исполнен, расторгнут и др. Такой подход позволяет быстро оценивать риски несвоевременного выполнения ПСД и степень завершённости проектных работ.

**4. Отслеживать динамику ПСД по годам**

График показывает изменение количества заключённых договоров ПСД по годам, что помогает анализировать тенденции роста проектной нагрузки,

Алматы қаласы цифрландыру басқармасы Адәндібай А. Ә.

активность закупок и бюджетные циклы.

**5. Просматривать детализированные договоры ПСД**

В таблице представлены ключевые данные по каждому договору: заказчик, финансовый год, номер договора, статус и подробное описание выполняемых проектных работ. Детализация позволяет оперативно анализировать конкретные объекты и стадии их реализации.

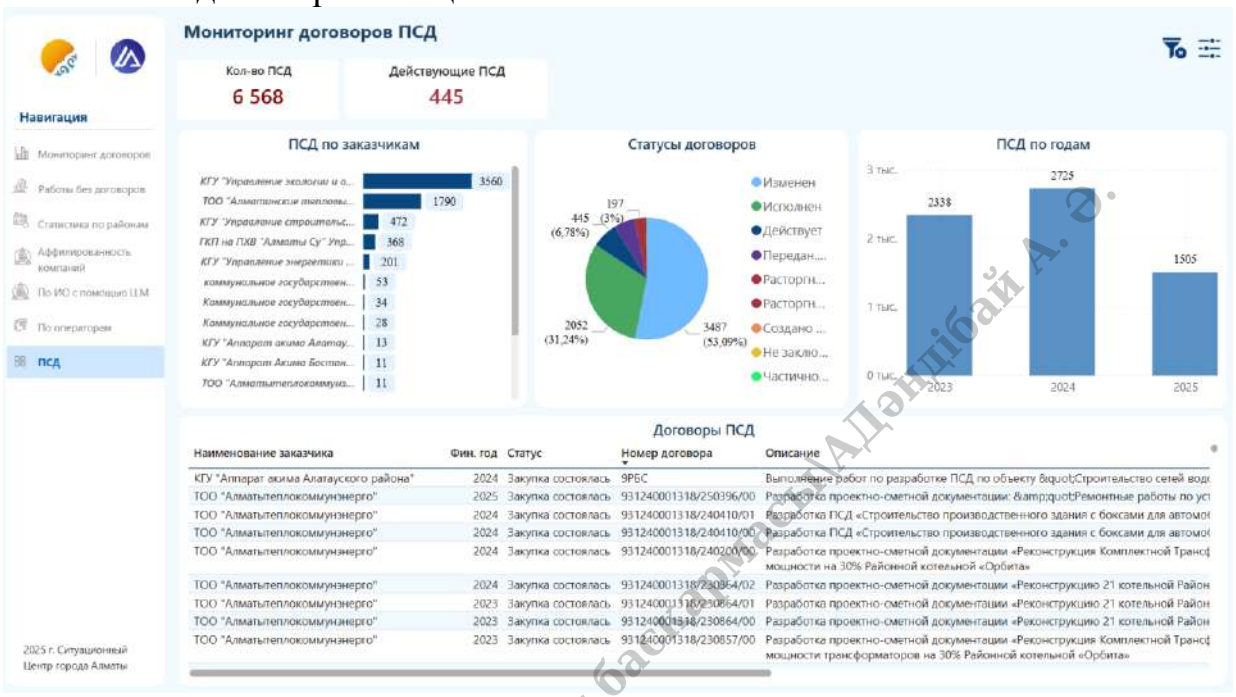



Рисунок – 2.3.4.14. Дашборд по мониторингу договоров ПСД

## 2.7.Разработка сервиса «Карта туристических объектов города Алматы»

***Цель:***

Создать систему для отображения и анализа туристических объектов города Алматы с целью управления туристическими потоками, повышения качества обслуживания и принятия обоснованных решений.

***Задачи:***

4-й квартал:

- Внедрение функции формирования отчетов по параметрам, включая популярность объектов, сезонность посещений и предпочтения туристов.
- Тестирование и запуск системы для пользователей.

***Источник данных:*** Управление туризма города Алматы, данные общественных и частных туристических объектов.

***Заказчик:*** Проектный офис ГРБН "Устойчивый экономический рост", Управление туризма.

Разработанный сервис доступен по ссылке: https://go.smartalmaty.kz/

Алматы қаласы цифрландыру басқармасы/Адаңбай А. Ә.

### 2.7.1. Внедрение функции формирования отчетов по параметрам, включая популярность объектов, сезонность посещений и предпочтения туристов

В рамках задачи четвертого квартала был разработан и внедрен аналитический модуль «GoAlmaty», отображающий ключевые показатели туристической активности в городе Алматы.

На панели формируется сводный аналитический отчет, который включает следующие элементы:

- *Популярный сезон посещений*, определенный по количеству созданных маршрутов.
- *Количество туристов*, которые сформировали маршрут через сервис.

Далее представлены визуальные диаграммы, отражающие структуру интересов туристов:

- *Популярные категории маршрутов* (история и культура, ночная жизнь, гастрономия и т.д.).
- *Популярные туристические объекты*, ранжированные по числу посещений.

Отдельный блок отображает информацию по предпочтениям туристов по дням недели — какой вид досуга оказывается наиболее востребованным в тот или иной день.

На интерактивной карте отмечены туристические точки, что позволяет визуально оценить концентрацию активности и распределение интереса по городу.

Также в модуле отображены два аналитических графика:

- Посещения по районам, где можно увидеть, какие районы города наиболее популярны среди туристов.
- График сезонности, демонстрирующий динамику посещений в течение осени и зимы с указанием пиковых недель.

Алматы қаласы цифрландыру басқармасы\АДендібай А.Э.

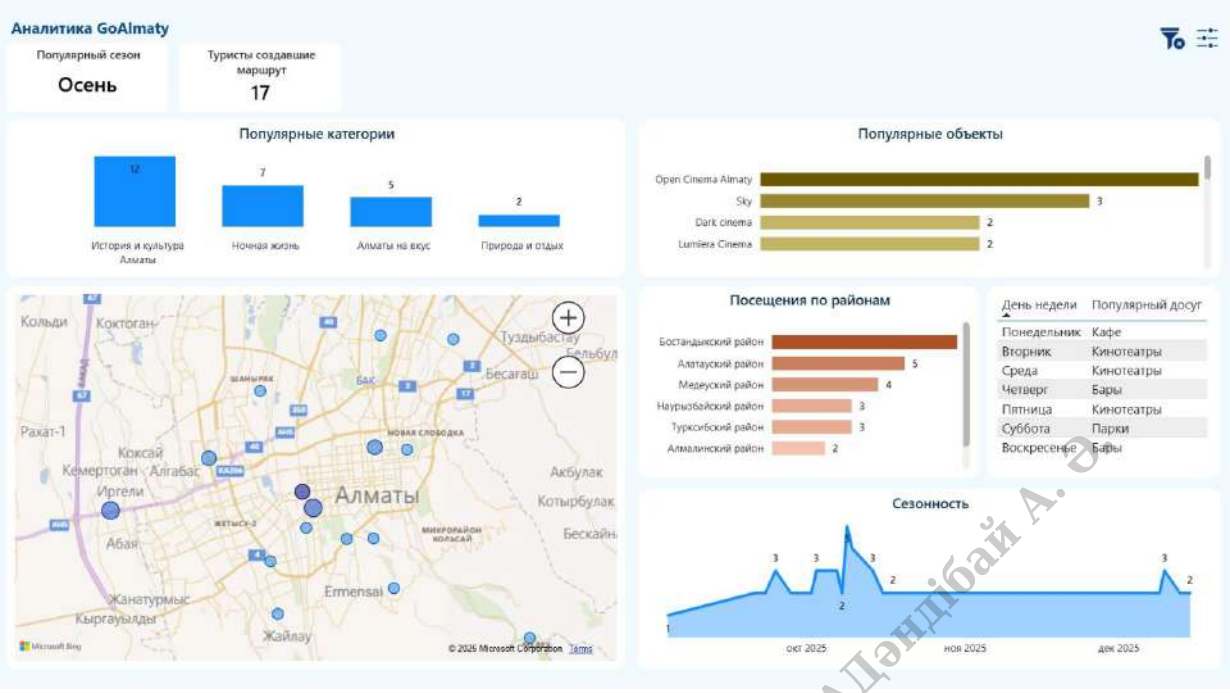



Рисунок - 2.3.5.1. Отображение загруженных данных

### 2.7.2. Тестирование и запуск системы для пользователей

В рамках подготовки сервиса «Карта туристических объектов города Алматы» к промышленной эксплуатации был выполнен полный цикл тестирования и процедур запуска, включающий проверку функциональности, производительности, корректности данных, интерфейса и устойчивости работы под нагрузкой. По завершении тестирования сервис был успешно развернут и предоставлен пользователям в составе цифровой экосистемы города.

1) Функциональное тестирование

Проведена проверка всех ключевых функций сервиса, включая:

- Работа интерактивной карты
  - корректность отображения базовой картографической подложки;
  - загрузка и визуализация туристических объектов различных категорий (достопримечательности, музеи, маршруты, парки, культурные площадки, сервисы и т.д.);
  - корректность геолокационного позиционирования объектов;
  - стабильность работы при переключении слоёв и фильтров.
- Отображение и структурирование карточек объектов
  - корректная подгрузка описаний, фотографий, классификаций, режимов работы, стоимости и других атрибутов;
  - проверка мультиязычности карточек (RU/KZ/EN);
  - переход к внешним ресурсам (сайты, соцсети, билеты).
- Работа механизмов поиска и фильтрации
  - фильтрация по категориям, популярности, близости, историческим периодам;

Алматы қаласы цифрландыру басқармасы АДәндібай А. е

- обработка поисковых запросов по названию объекта;
- проверка корректности ранжирования результатов.

– Маршруты и навигация
  - построение туристических маршрутов;
  - расчёт времени переходов;
  - корректность отображения рекомендаций.

2) Тестирование данных

Выполнена проверка полноты и достоверности входящих данных:

– географические координаты объектов;
– соответствие категорий;
– корректность описаний и метаданных;
– качество мультимедийных материалов;
– валидация отсутствия дублирующихся объектов;
– согласованность данных между исходными системами и витриной сервиса.

Для выявленных расхождений выполнена корректировка данных и восстановление несопоставимых записей.

3) Тестирование интерфейса и UX

Проведены проверки пользовательского опыта:

– корректность отображения на мобильных устройствах, планшетах и десктопах;
– отзывчивость интерфейса и скорость интеракций;
– доступность ключевых функций;
– удобство взаимодействия с картой и карточками объектов;
– адаптация для разных языковых версий.

По итогам были внесены UX-улучшения: уточнены подписи, оптимизирована структура карточек, ускорена загрузка изображений.

4) Интеграционные тесты

Проверена работа интеграций с:

– модулем геоаналитики и картографического рендеринга;
– внешними источниками данных туристической инфраструктуры;
– сервисом маршрутизации;
– системой обновления данных;
– API получения справочной информации.

Все каналы синхронизации подтвердили корректность работы.

5) Подготовка среды, запуск и публикация сервиса

После завершения всех испытаний выполнены:

– Развертывание сервиса
  - перенос решения в производственную инфраструктуру;
  - настройка доменного имени и сервиса публикации;
  - включение кэширования картографических данных;
  - активация системы мониторинга и журналирования событий.

Алматы қаласы Цифрлық даму басқармасы\Әлнәбай А. Ә.

- Настройка прав доступа
  - создание пользовательских ролей;
  - определение прав на просмотр, редактирование и загрузку данных;
  - подключение административной панели при необходимости.
- Финальный контроль качества
  - проверка работоспособности всех функций на продакшене;
  - тестирование на нескольких устройствах;
  - контроль соответствия визуального оформления дизайн-коду города Алматы.
- Публикация сервиса для пользователей
  Сервис был официально открыт для использования:
  - добавлен в цифровую экосистему и каталог сервисов;
  - доступен через веб-интерфейс https://go.smartalmaty.kz/;
  - сопровождается пользовательскими инструкциями и справочными материалами.

**2.8.Разработка сервиса «Цифровой рейтинг школ».**

***Цель:***

Разработать и внедрить цифровую платформу для объективной оценки и ранжирования общеобразовательных школ города Алматы с целью повышения качества образовательных услуг.

***Задачи:***

4-й квартал:

- Проведение пилотного запуска сервиса на стороне Заказчика.
- Обучение пользователей работе с системой, с предоставлением обучающих материалов, и сбор обратной связи.

***Источник данных:*** Информационные системы образовательных учреждений, электронные журналы, системы управления обучением.

***Заказчик:*** Проектный офис ГРБН «Инклюзивная среда, образование и медицина», Управление образования города Алматы.

**2.8.1. Проведение пилотного запуска сервиса на стороне Заказчик**

В рамках внедрения сервиса «Цифровой рейтинг школ» была проведена комплексная процедура пилотного запуска модуля на стороне Заказчика, включающая техническую интеграцию, проверку корректности данных и организацию доступа для пользователей. Пилотная версия официально развернута и доступна по адресу:

https://schools-ui.vercel.app/ru/schools/rating

Работы выполнялись поэтапно:

1) Подготовка инфраструктуры и среды размещения

Были выполнены работы по настройке серверной среды и окружения, включая:

Алматы қаласы цифрландыру басқармасы\Адәнбай А. Ә.

- размещение фронтенд-части сервиса на платформе Vercel с обеспечением автоматических билдов и деплоя;
- конфигурацию API-шлюза для взаимодействия с данными Ситуационного центра;
- настройку защищённых соединений и контроль доступа к сервису через тестовые токены;
- интеграцию с системой логирования и мониторинга производительности.

2) Загрузка, проверка и нормализация данных

Для обеспечения корректной работы алгоритмов рейтинга произведена загрузка исходных данных.

Выполнены:

- валидация заполненности и форматов данных;
- нормализация и приведение атрибутов к требуемым шкалам;
- проверка корректности расчётов рейтинговых баллов;
- тестовые прогоны модели на выборочных школах для оценки распределения баллов.

3) Функциональное тестирование сервиса

Проведён набор тестов, направленных на проверку корректной работы пользовательского интерфейса и логики расчётов:

- тестирование фильтров (район, категория, тип школы);
- проверка сортировок, пагинации, отображения карточек школ;
- валидация корректности отображения рейтинговых значений;
- проверка мультиязычного интерфейса (RU/KZ);
- тестирование сценариев поведения на различных устройствах (десктоп, планшет, мобильный).

4) Тестирование алгоритмов рейтинга и аналитической части

Проведена оценка корректности работы модели расчёта рейтинга, включающая:

- проверку весовых коэффициентов и итоговых формул;
- сравнение результатов с эталонными статистическими выборками;
- анализ распределения баллов для исключения аномалий (завышенных/заниженных значений);
- тестирование обновления данных и повторного пересчёта рейтинга.

5) Интеграционные испытания

Для оценки стабильности работы сервиса выполнены:

- тестирование скорости отклика API и времени рендера страниц;
- проверка отказоустойчивости и восстановления после ошибок;
- контроль корректности взаимодействия с внешними системами данных.

6) Организация тестового доступа для Заказчика

Алматы қаласы цифрландыру басқармасы\Адилбай А. Ә.

Проведены действия по предоставлению доступа к пилотной версии:

- создание тестовых аккаунтов и ролей пользователей;
- настройка уровней доступа для ответственных сотрудников Управления образования;
- предоставление инструкций по использованию пилотной версии.

7) Сбор обратной связи и корректировки

После публикации пилотной версии был организован процесс получения комментариев от представителей Заказчика.

Выполнено:

- сбор замечаний и предложений;
- классификация вопросов по функциональным блокам;
- внесение корректировок в интерфейс, фильтры, отображение данных и логику расчётов;
- обновление пилотной версии с учетом замечаний

### 2.8.2. Обучение пользователей работе с системой, с предоставлением обучающих материалов, и сбор обратной связи

Для обеспечения эффективного использования сервиса «Цифровой рейтинг школ» ответственными сотрудниками Заказчика была организована комплексная программа обучения пользователей, включающая проведение обучающих сессий, подготовку методических материалов и сбор обратной связи по результатам эксплуатации пилотной версии.

1) Проведение обучающих сессий

Были организованы обучающие мероприятия с участием пользователей из Управления образования г. Алматы и специалистов Проектного офиса ЦРА. В рамках обучения были детально рассмотрены:

- структура сервиса и ключевые функциональные блоки;
- правила навигации и использования интерфейса;
- работа с фильтрами, рейтингами, карточками школ и аналитическими элементами;
- процедура обновления данных и сценарии использования сервиса для управленческих решений;
- интерпретация рейтинговых показателей и аналитических графов.

Сессии проводились в онлайн- и очном форматах, включая демонстрацию функционала на реальных данных и пошаговые инструкции по выполнению основных операций.

2) Подготовка обучающих и методических материалов

Пользователям были предоставлены специализированные материалы, в том числе:

- руководство пользователя с пошаговыми описаниями функций;
- презентационные материалы для самостоятельного изучения;
- видеоролики с демонстрацией ключевых сценариев работы;

Алматы қаласы цифрландыру басқармасы(Аденбай А. Ә.

- методические пояснения по логике формирования рейтингов и принципам расчёта;
- FAQ-набор по наиболее распространенным вопросам.

Материалы подготовлены в многоязычном формате (RU/KZ) и доступны в электронном виде.

3) Консультации и сопровождение пользователей

В рамках пилотного периода обеспечивалась поддержка пользователей:

- оперативные консультации по работе с системой;
- помощь в интерпретации аналитических показателей;
- сопровождение при выявлении технических затруднений;
- предоставление разъяснений по алгоритмам расчета рейтингов.

Организована коммуникационная линия для приёма обращений (почтовый канал, мессенджеры, форма обратной связи).

4) Сбор обратной связи и анализ предложений

Для улучшения качества сервиса была проведена системная работа по сбору обратной связи:

- опросы пользователей по функциональности и удобству интерфейса;
- фиксация замечаний, пожеланий и выявленных несоответствий;
- анализ отзывов по группам: интерфейс, функциональность, данные, производительность, аналитические элементы.

На основе обратной связи подготовлен перечень приоритетных доработок, часть из которых была реализована в рамках пилотной стадии, включая улучшение отображения карточек школ, уточнение фильтров, оптимизацию скорости загрузки данных и корректировку отдельных формул расчета.

5) Актуализация обучающих материалов

После внесения доработок обучающие материалы были обновлены и повторно предоставлены пользователям, включая:

- обновленные инструкции;
- скорректированные видеодемонстрации;
- расширенный раздел FAQ.

### 2.9.Разработка модуля «Мониторинг госпитализаций и коечной мощности»

***Цель:***

Обеспечить контроль за госпитализациями и эффективное управление коечной мощностью стационаров города.

***Задачи:***

4-й квартал:

- Разработка рекомендаций по улучшению управления коечной мощностью.

***Результаты:***

- Карта загруженности стационаров с указанием коечной мощности.

Алматы қаласы цифрлық даму басқармасы\Адильбай А.Э.

– Отчет о динамике госпитализаций и отказов.
– Рекомендации по оптимизации управления стационарами.

***Источник данных:*** Медицинские учреждения города Алматы.

***Заказчик:*** Проектный офис ГРБН «Инклюзивная среда, образование и медицина», Управление здравоохранения города Алматы.

### 2.9.1. Анализ данных

По направлению «Здравоохранение» ранее был разработан дашборд, состоящий из трёх страниц:

– Главная;
– Инфраструктура;
– Персонал.

Главная страница содержит ключевые показатели и интерактивную карту с координатами поликлиник, где при выборе объекта отображается территория обслуживания и детальная информация.

На странице «Инфраструктура» реализована аналитика с диаграммами и таблицей, отображающей сведения о зданиях и индексе нагрузки.

Страница «Персонал» включает таблицу «Анализ по районам» и диаграмму «Обеспеченность врачами», а также таблицу нагрузки на врачей с цветовой индикацией по уровню нагрузки.

В текущем месяце проведена работа по улучшению функционала и дизайна дашборда, а также внедрению методологии, направленной на улучшение управления коечной мощностью.

В части внедрения методологии, добавлена поп-ап страница, подробно описывающая, как рассчитываются все показатели дашборда.

В методологии указаны используемые формулы и логика вычислений. Для удобства отображения методология реализована во всплывающем окне (pop-up).

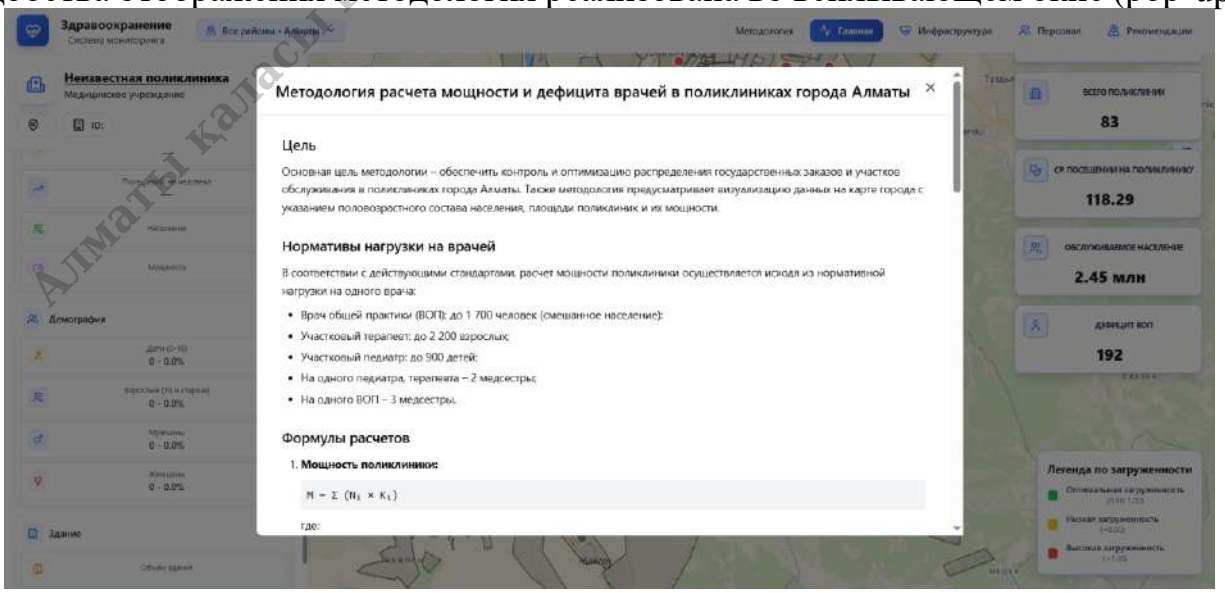



Рисунок – 2.3.7.1. Методология для управления коечной мощностью

Алматы қаласы инфрақұрылым басқармасы Әдәмбай А. Ә.

При разработке методологии были использованы действующие стандарты по нагрузке на врачей.

В соответствии с действующими стандартами, расчет мощности осуществляется исходя из нормативной нагрузки на 1 врача:

- Врач общей практики – до 1 700 человек;
- Участковый терапевт – до 2 200 взрослых;
- Участковый педиатр – до 900 детей;
- На одного педиатра, терапевта – 2 медсестры;
- На одного ВОП – 3 медсестры.

Реализована фильтрация по районам для карты: при выборе района отображаются только соответствующие поликлиники и их обслуживаемые территории. (Рисунок – 2.3.7.2)

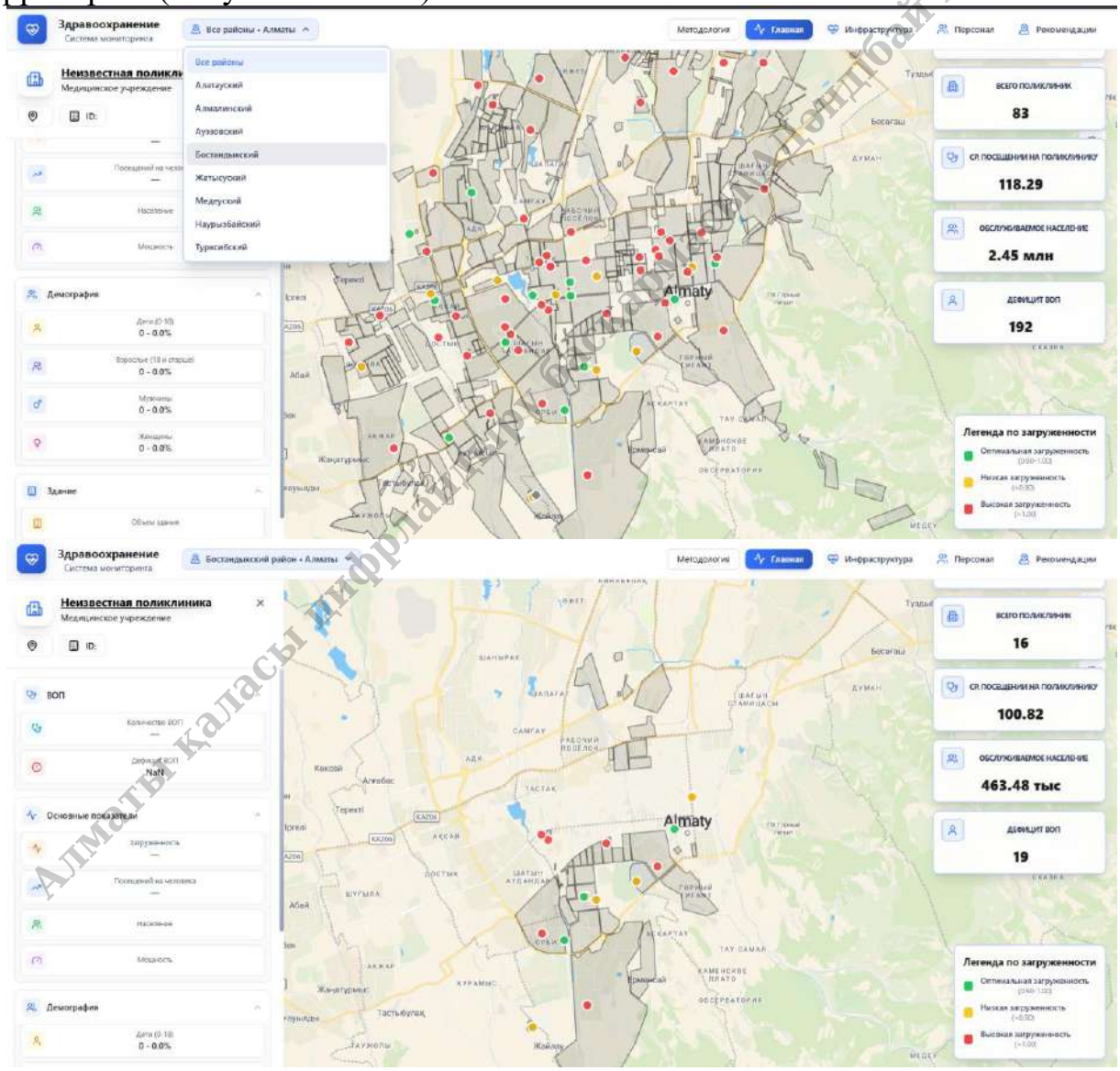



Рисунок – 2.3.7.2. Функция фильтрации по районам

Также проведены работы по улучшению дизайна дашборда в соответствии с обновлённой версией макета в Figma.

Алматы каласы цифрлайдыр бастарыны Мэндібай А. Э.

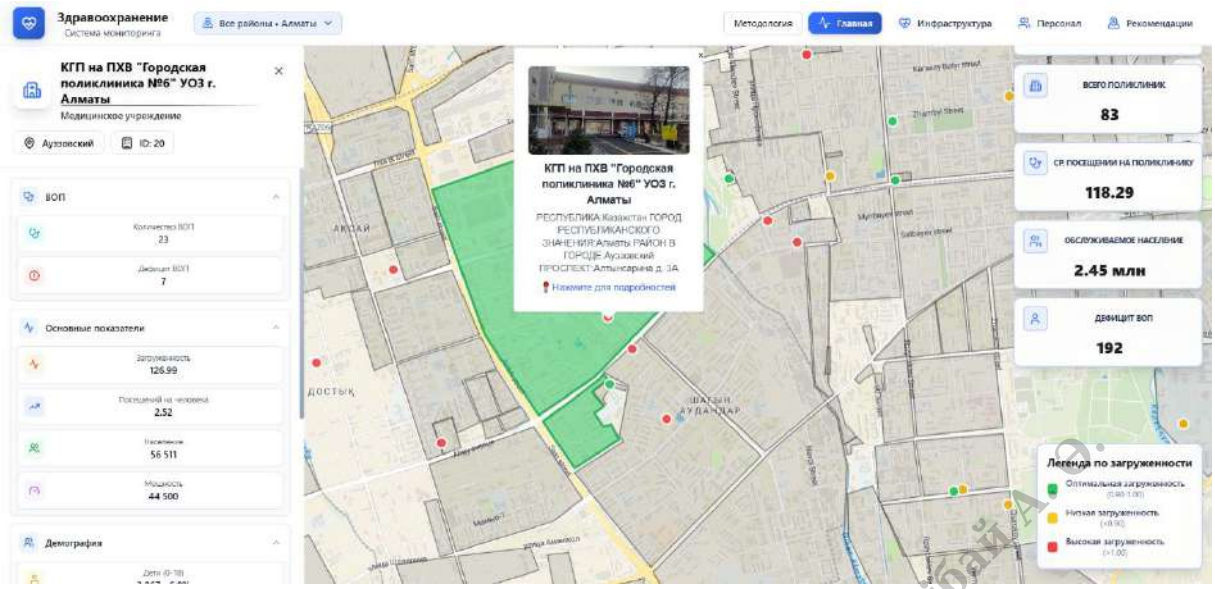



Рисунок – 2.3.7.3. Новый дизайн

На странице «Инфраструктура» добавлена интерактивность между элементами визуализации — таблицы, гистограммы и диаграммы теперь связаны между собой.

При выборе элемента на одной из гистограмм автоматически фильтруются данные на остальных визуализациях. (Рисунок – 2.3.7.4)

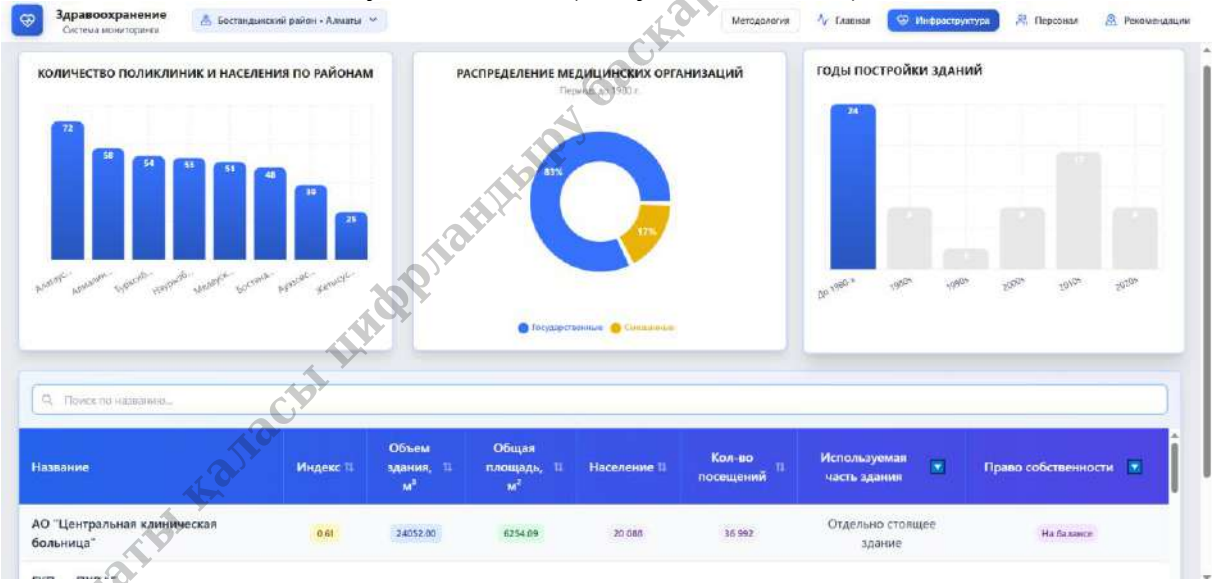



Рисунок – 2.3.7.4. Интерактивность между элементами визуализации

На странице «Персонал» произведено улучшение дизайна для визуальной согласованности со всеми страницами дашборда. (Рисунок – 2.3.7.5)

Алматы қаласы цифрландыру басқармасы | Дандыбай А.Ә.

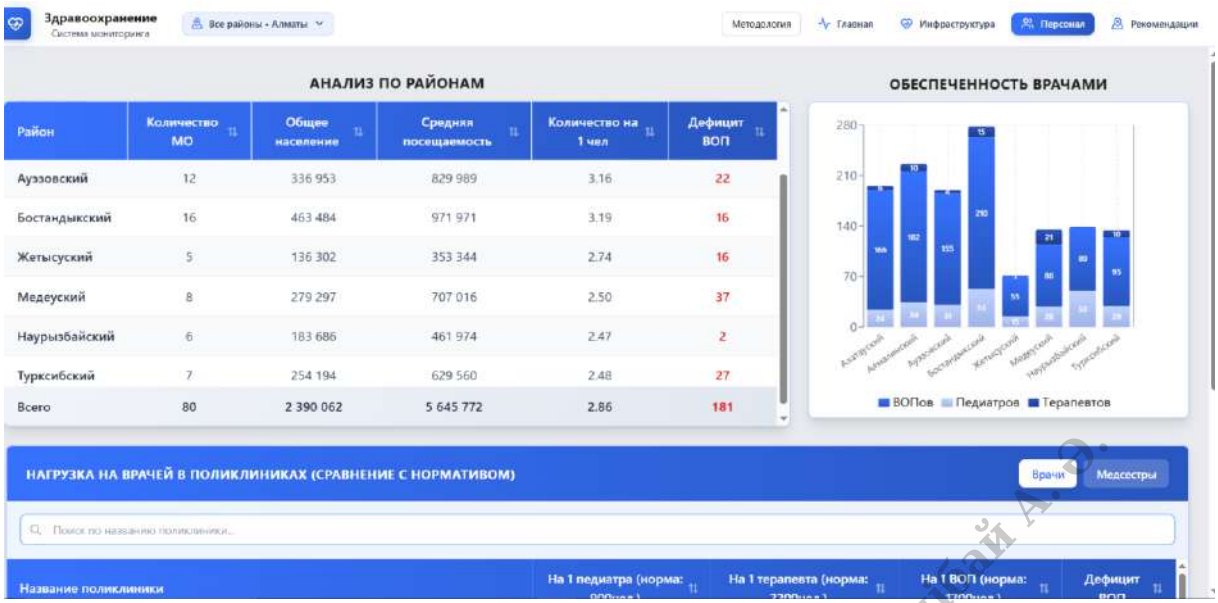



Рисунок – 2.3.7.5. Дашборд «Персонал»

Добавлены функции сортировки в таблице (по возрастанию и убыванию). (Рисунок – 2.3.7.6)

Рисунок – 2.3.7.6. Функция сортировки данных

Исправлены ошибки в таблице, добавлена цветовая индикация значений в зависимости от показателей. (Рисунок – 2.3.7.7)

Рисунок – 2.3.7.7. Таблица

В таблицу добавлена новая колонка «Дефицит ВОП», отражающая нехватку врачей общей практики по каждой поликлинике.

В таблицу добавлен новый раздел «Медсестры», отображающий количество медсестёр в каждой поликлинике. (Рисунок – 2.3.7.8)

Алматы қаласы цифрландыру басқармасы\Арапбай А. Э.

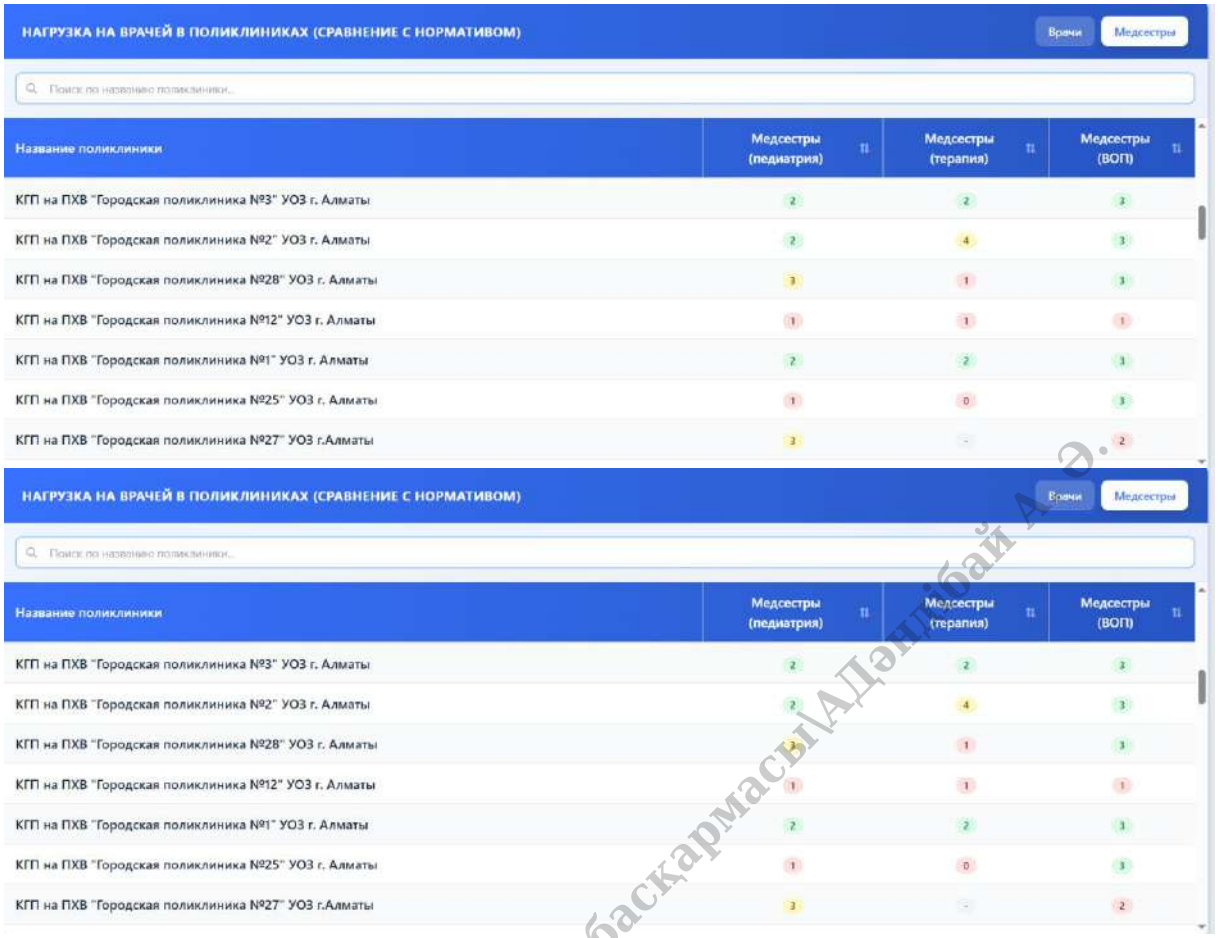



Рисунок – 2.3.7.8. Новые таблицы

Реализован поиск по наименованию поликлиники. (Рисунок – 2.3.7.9)

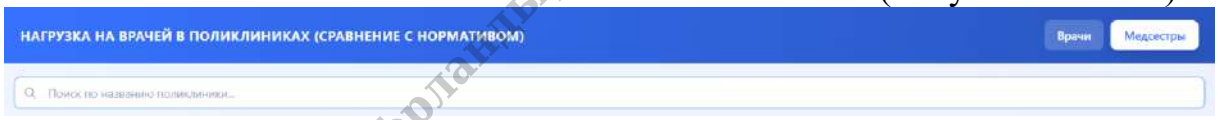



Рисунок – 2.3.7.9. Функция поиска данных

### 2.9.2. Разработка рекомендаций по улучшению управления коечной мощностью

Рекомендации по улучшению управления кочной мощносью был разработан для оперативного контроля нагрузки стационаровгорода Алматы, анализа доступности больниц, прогнозирования потребности в койках и выдачи автоматических рекомендаций по оптимизации.

Рекомендации включают:

- анализ загруженности коечного фонда по каждой МО;
- рекомендации по перенаправлению пациентов;
- расчёт недостающего количества коек;
- определение ближайших больниц по профилю и расстоянию;
- дополнительный блок для СМП (скорой медицинской помощи);
- анализ транспортной доступности по дорожной сети.

Логика расчета расстояния между МО

Алматы қаласының цифрландыру басқармасы\АДәнібай А. Э.

Для оценки доступности ближайших больниц используется Haversine формула, учитывающая кривизну Земли.

**Математическая формула**

$$a = \sin^2(\Delta\phi/2) + \cos(\phi_1) \times \cos(\phi_2) \times \sin^2(\Delta\lambda/2)$$
$$c = 2 \times \operatorname{atan2}(\sqrt{a}, \sqrt{(1-a)})$$
$$d = R \times c$$

**Где:**

- $\phi$ широта (latitude) в радианах
- $\lambda$ долгота (longitude) в радианах
- $R$ радиус Земли (6371 км)
- $\Delta\phi$ разница широт ($\varphi_2 - \varphi_1$)
- $\Delta\lambda$ разница долгот ($\lambda_2 - \lambda_1$)
- $d$ итоговое расстояние в км

***Логика расчета расстояния***

- Координаты больниц переводятся из градусов в радианы.
- Высчитывается разница широт и долгот.
- С помощью тригонометрии учитывается кривизна Земли.
- Вычисляется центральный угол между точками.
- Угол умножается на радиус Земли → получаем расстояние в километрах.

***Расчет ориентировочного времени прибытия СМП***

$$\text{время (мин)} = \frac{d}{40\ \text{км/ч}} \times 60$$

Используется средняя скорость города, учет пробок и светофоры.

Ограничения метода

- расстояние считается по прямой, не по реальным дорогам;
- время ориентировочное;
- для маршрутов рекомендуется 2ГИС/Яндекс.

***Справочник профилей и распределение больниц***

Создан единый классификатор профилей, который позволяет:

- осуществлять поиск больниц только по релевантным профилям (кардиология, онкология, гинекология и др.);
- корректно подбирать альтернативные медицинские организации для перенаправления;
- исключать учреждения, не предоставляющие требуемые виды услуг.

Справочник используется в следующих целях:

- для рекомендаций по перенаправлению пациентов;
- в блоке скорой медицинской помощи (СМП);
- при визуализации профилей перегрузки медицинских организаций.

***Рекомендации по перенаправлению пациентов***

Логика работы алгоритма

Алматы қаласы цифрландыру басқармасы Әдәлібай А. Ә.

1) Определяем МО, чья загруженность превышает порог:
   - 85% — высокая нагрузка
   - 95% — критическая
2) Учитываем профиль (например, "Онкология", "Гинекология", "Кардиология").
3) Находим альтернативные больницы, которые:
   - имеют тот же профиль,
   - находятся в радиусе, например, до 10 км,
   - имеют свободные койки.
4) Если альтернативы есть → они выводятся в списке с указанием:
   - расстояния,
   - времени в пути,
   - фактической загруженности.
5) Если альтернатив нет → рассчитываем требуемое количество дополнительных коек.

***Формула расчёта оптимального количества коек***

Когда больница перегружена, система считает, сколько коек нужно добавить, чтобы вернуть загрузку к оптимальному значению (например, ≤ 85%).

Пусть:

- CCC — фактическое количество коек
- LLL — текущая загрузка (например, 118%)

Сначала вычисляем фактическое количество пациентов:

P=C·L100P = C \cdot \frac{L}{100}P=C·100L

Далее определяем оптимальное количество коек:

Coptimal=P0.85C_{\text{optimal}} = \frac{P}{0.85}Coptimal=0.85P

Тогда:

Дополнительные койки=Coptimal−C\text{Дополнительные койки} = C_{\text{optimal}} - CДополнительные койки=Coptimal−C

**Пример**

- Больница имеет 25 коек
- Загрузка = 118%

Пациентов 25·1.18=3025 \cdot 1.18 = 3025·1.18=30

Оптимально нужно:

Coptimal=30/0.85≈35C_{\text{optimal}} = 30 / 0.85 \approx 35Coptimal =30/0.85≈35

Недостача:

35−25=10 коек35 - 25 = 10 \text{ коек}35−25=10 коек

Эта логика встроена в рекомендации, отображаемые на карте и в карточках МО.

Алматы қаласы цифрлаңдыру басқармасы\АДәндібай А. Ә.

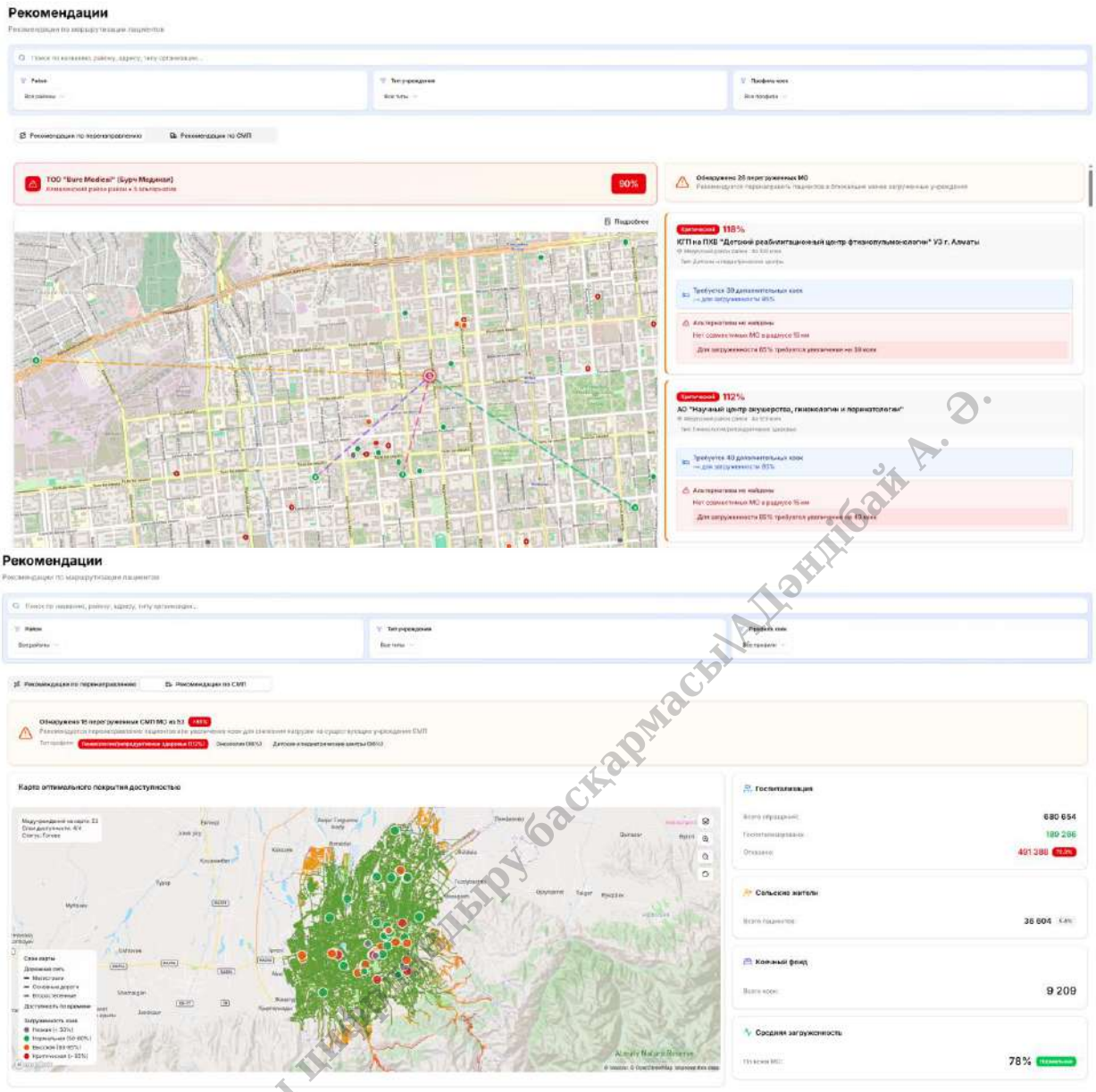



Рисунок – 2.3.7.10. Рекомендации

Алматы қаласы цифрландыру басқармасы/Дәндікай А. Ә.

**2.10. Разработка модуля «Карта нозологий и смертности».**

**Цель**: Обеспечить мониторинг заболеваемости и смертности для принятия эффективных решений по профилактике и лечению.

**Задачи**:

4-й квартал:

- Реализовать обновления данных на постоянной основе посредствам интеграции (при наличии возможности со стороны сторонних ИС).
- Визуализация данных на карте города с детализацией по участкам ВОП.
- Разработка визуализации данных, превышающих городские и Республиканские показатели.

**Результаты**:

- Карта заболеваемости и смертности с детализацией по районам.

– Отчеты с рекомендациями для улучшения качества медицинских услуг.

**Источник данных**: Управление здравоохранения, медицинские учреждения города Алматы.

**Заказчик**: Проектный офис ГРБН «Инклюзивная среда, образование и медицина», Управление здравоохранения города Алматы.

По причине задержки вопроса интеграции со стороны Министерства здравоохранения, Ситуационный центр не имеется данных о заболеваемости (данные не предоставляются в виде выгрузки). Интеграция согласована, техническая реализация запланирована на октябрь 2025 года.

По согласованию с заказчиком реализация задач 3 квартала будет производиться в 4 квартале по факту получения данных со стороны ИС МЗ РК.

### 2.10.1. Реализовать обновления данных на постоянной основе посредствам интеграции

В рамках выполнения задач 4-квартала, были выполнены работы по разработки API, позволяющей обновлять данные на постоянной основе. Также проведены комплексные работы на стороне backend, позволяющий производить аналитику больших данных.

Разработаны ViewSets и Serializers для работы со справочниками причин смерти, диагнозов и медицинских учреждений.

Разработаны следующие алгоритмы автоматизированной обработки данных о смертности для формирования отчетов в реальном времени:

***Группировка по МКБ-10***

Написан алгоритм (stat_by_mkb_group), который автоматически распределяет коды диагнозов по 16 основным классам (Инфекционные, Новообразования, БСК и др.).

Реализована группировка буквенно-цифровых кодов для корректного попадания в диапазоны заболеваний.

***Оценка материнской и младенческой смертности***

Разработан метод определения зоны ответственности (_calculate_maternal_responsibility). Система анализирует диагноз и определяет уровень летального исхода: поликлиника (косвенные причины) или стационар (прямые акушерские причины).

Внедрен расчет коэффициентов смертности на 1000 родившихся с использованием исторических данных по рождаемости за 2021–2025 годы.

### 2.10.2. Визуализация данных на карте города с детализацией по участкам ВОП

Разработан и внедрен модуль «Система мониторинга карты нозологий и смертности», состоящий из двух страниц:

– Главная страница;

Алматы қаласы Цифрландыру басқармасы/Дәндібай Н. Ә.

– Аналитика.

***Главная страница***

На карте отображаются координаты медицинских учреждений (частные организации, поликлиники, центры/клиники, институты и др.) с наложением границ территорий районов города. (Рисунок – 2.3.8.1.)

В левом верхнем углу размещен панель-фильтр, позволяющая выбирать район и тем самым фильтровать отображение территории района и медучреждений внутри него. (Рисунок - 2.3.8.2)

В панели размещен главный показатель «Всего поликлиник».

При клике на маркер медицинского учреждения отображается всплывающее окно с краткой информацией и одновременно раскрывается дополнительная карточка с подробной информацией в нижней части панели фильтрации. (Рисунок - 2.3.8.3)

Для удобства просмотра карты, предусмотрен режим сворачивания панели фильтрации. (Рисунок – 2.3.8.4)

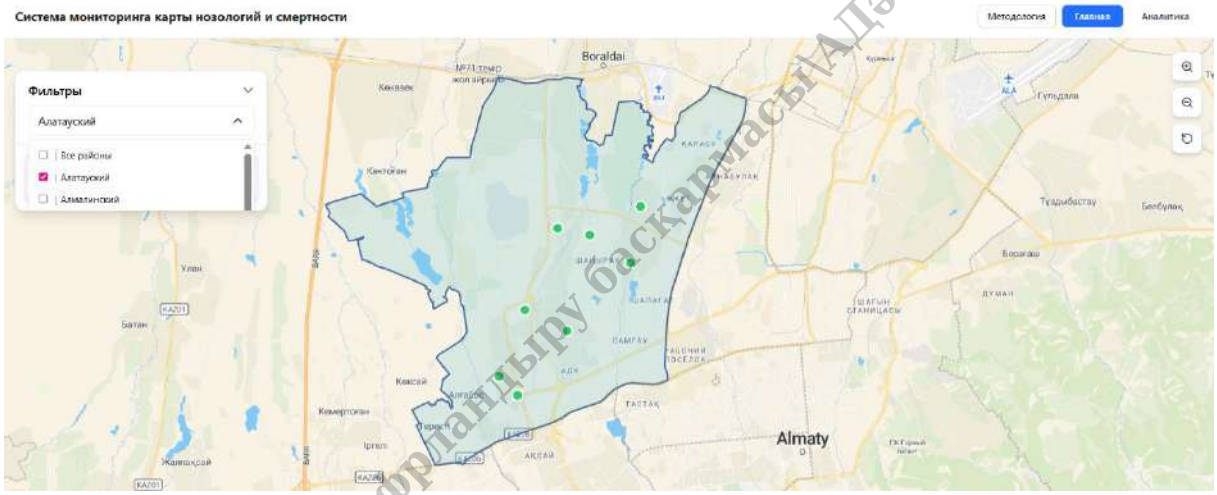



Рисунок – 2.3.8.1. Карта медицинских учреждений
с отображеним границ территории города

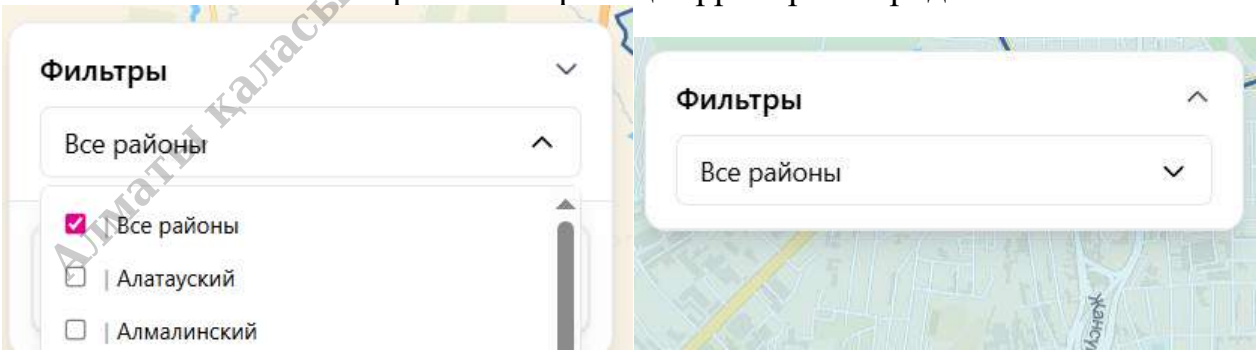



Рисунок – 2.3.8.2. Фильтры

Алматы қаласы цифрландыру басқармасы\Дәндібай А. Ә.

Рисунок – 2.3.8.3. Всплывающее окно с карточкой медицинского учреждения

***Аналитика***

Страница «Аналитика» разделена на три формальных подраздела, каждый из которых выполняет отдельную аналитическую задачу:

**Подраздел 1 — Общая аналитика**

– График «Смертность по годам и по месяцам» с возможностью просматривать информацию: «по годам» (данные 2021–2025 гг.), «по месяцам года» (выбор конкретного года для просмотра помесячных данных) и «по месяцам (в среднем)» (усреднённые помесячные значения за 2021–2025 гг.). (Рисунок – 2.3.8.5)

– Диаграмма «Распределение по полу» (мужчины/женщины) (Рисунок – 2.3.8.6)

– В нижней части раздела размещены три гистограммы: «Топ-10 МКБ-групп», «Топ-10 причин смерти» и «Топ-10 смертей по МО (медицинским организациям)» (Рисунок – 2.3.8.7)

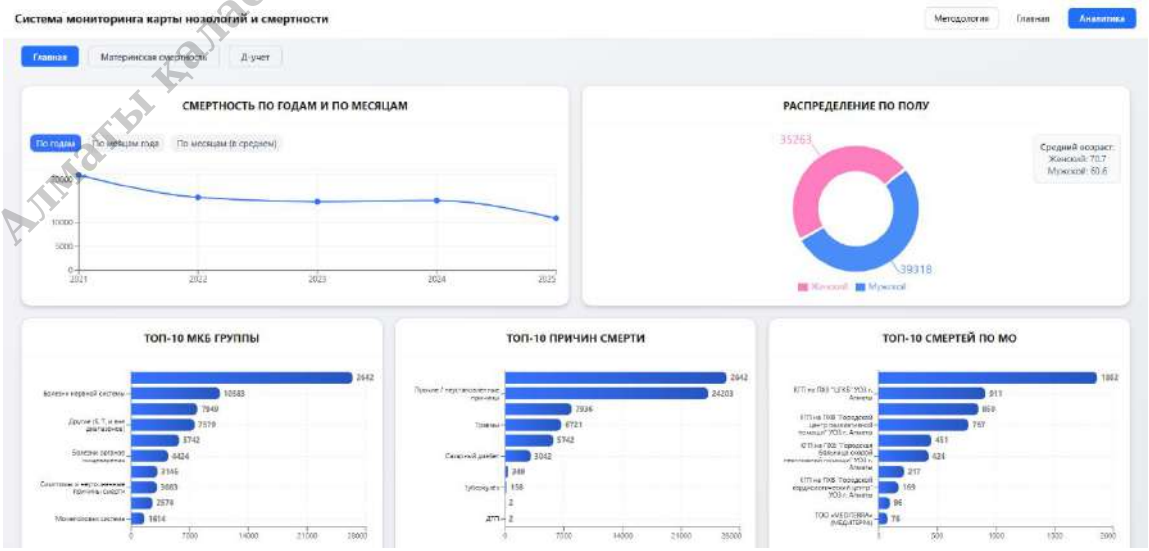



Рисунок – 2.3.8.4. Аналитика. Общая страница

Алматы қаласы цифрландыру басқармасы Аханділбай А. Ә.

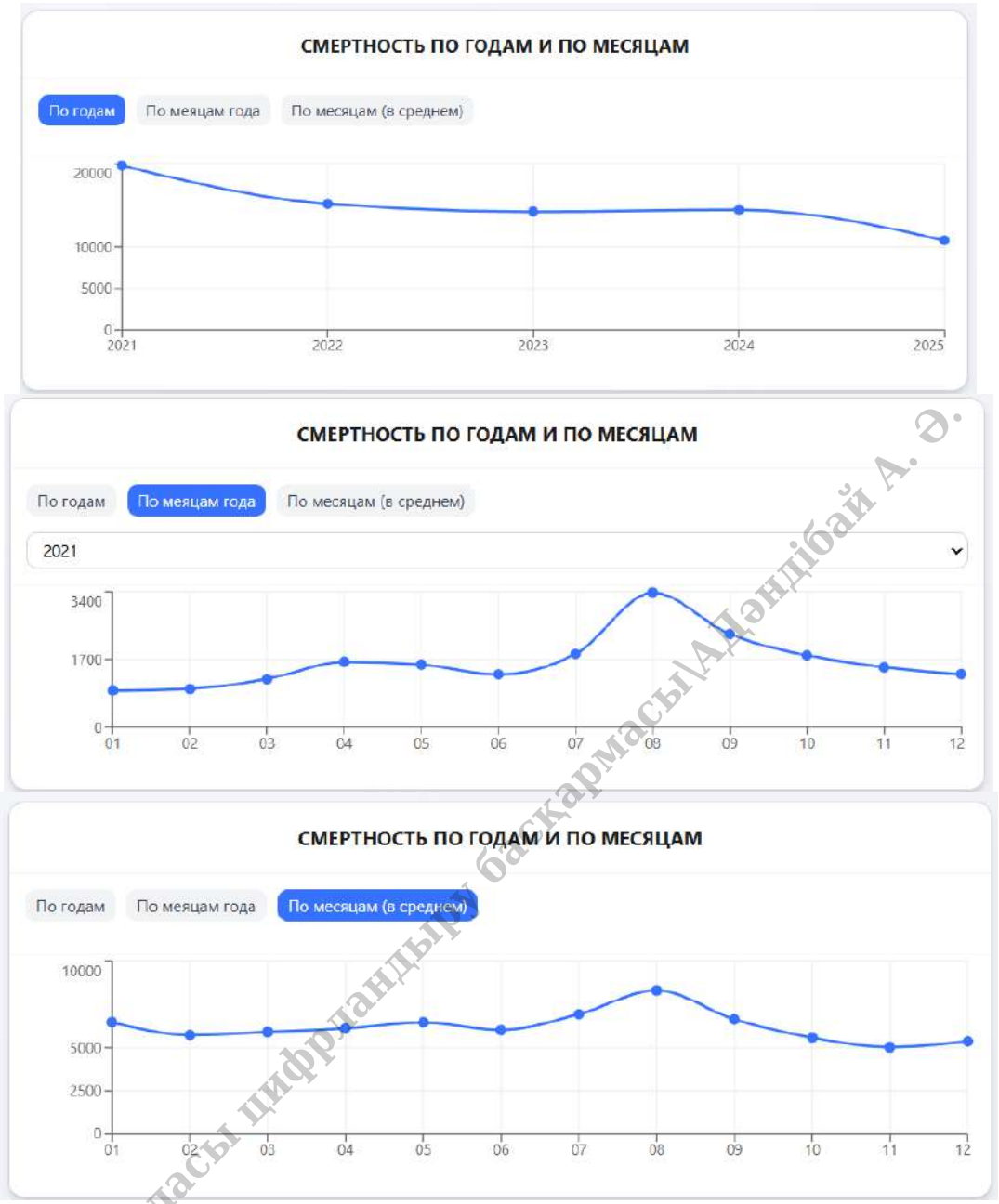



Рисунок - 2.3.8.5. График «Смертность по годам и по месяцам»

Рисунок – 2.3.8.6. Графика «Распределние по полу»

Алматы қаласы цифрландыру басқармасы\Аңәндібай А. Ә.

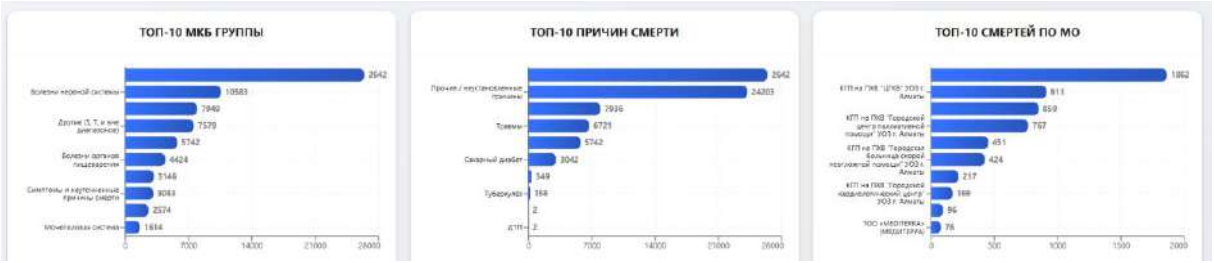



Рисунок – 2.3.8.7. Гистрограммы

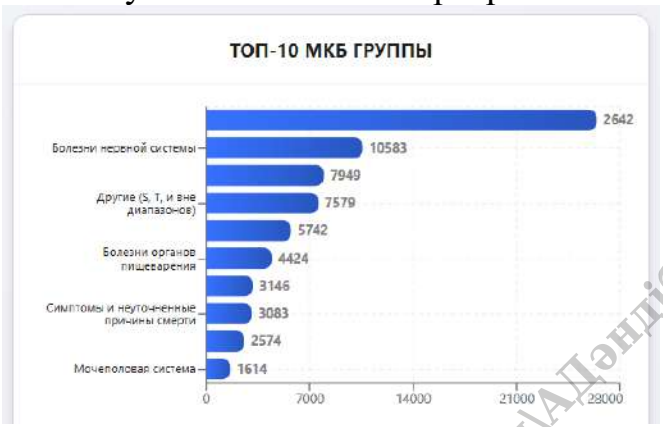



Рисунок – 2.3.8.8. Гистограмма «Топ-10 МКБ группы»

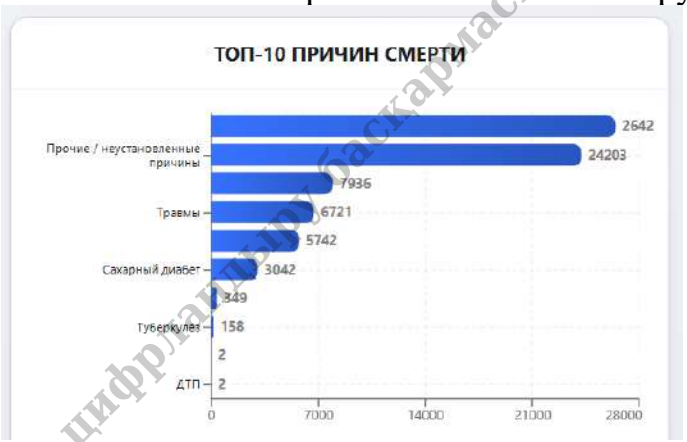



Рисунок – 2.3.8.9. Гистограмма «Топ-10 причин смерти»

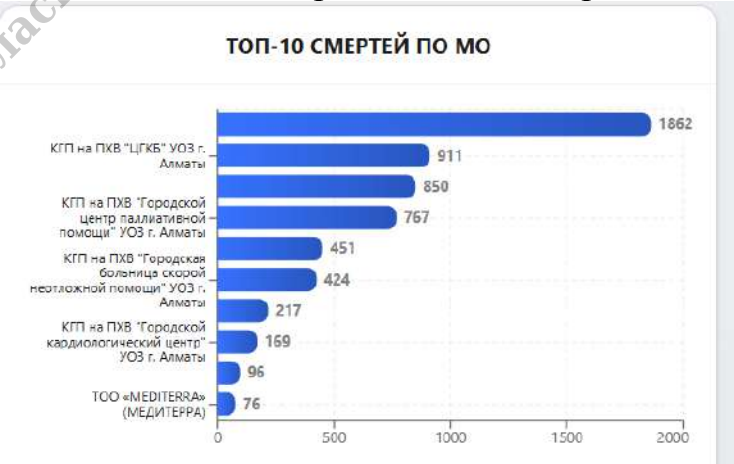



Рисунок – 2.3.8.10. Гистограмма «Топ-10 МО по сметрности»

**Подраздел 2 – Материнская смертность**

Дашборде «Материнская смертность» состоит из следующих

Алматы қаласы цифрландыру басқармасы\Әдәндібай А. Ә.

компонентов:

- График «Материнская смертность по годам и по месяцам» с переключением видов отображения («по годам» / «по месяцам»). (Рисунок – 2.3.8.11)
- Таблица «Материнская смерть по МО» — показатели материнской и детской смертности по родильным домам и соответствующим медицинским организациям. (Рисунок – 2.3.8.12)
- Таблица «Материнская смерть по прикреплённому МО» — показатели по прикреплённым организациям. (Рисунок – 2.3.8.12)
- Две круговые диаграммы: «Причины материнской смертности» и «Причины детской смертности». (Рисунок – 2.3.8.13)

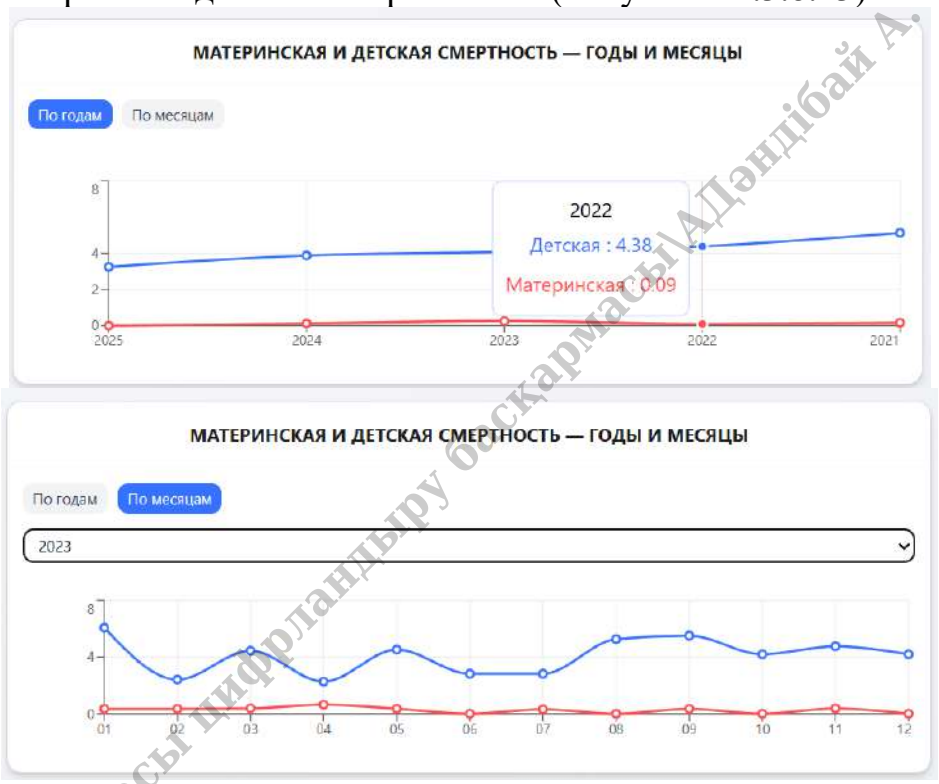



Рисунок – 2.3.8.11. Лайн чарт «Материнская и детская смертность»

Алматы қаласы цифрландыру басқармасы\АДәндібай А. Ә.

МАТЕРИНСКАЯ СМЕРТЬ ПО ПРИКРЕПЛЕННОМУ МО

| Название прикрепленного МО | Материнская смертность | Детская смертность |
|---|---|---|
| За городом Алматы | 6 | 23 |
| КГП на ПХВ "ГП №36" УОЗ г. Алматы | 1 | 3 |
| ТОО "Open medical channel" | 2 | 2 |

Рисунок – 2.3.8.12. Таблица «Материнская смертность по МО»

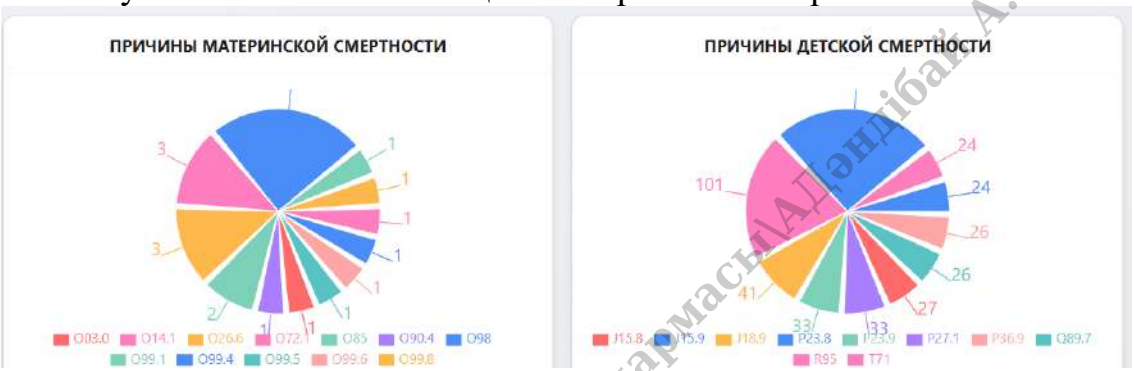



Рисунок – 2.3.8.13. Круговая диаграмма
«Причины материнской и детской смертности»

**Подраздел 3 – Д-учет**

Подраздел «Д-учет» состоит из следующей информации:

- Ключевые показатели: «Всего пациентов», «Средний возраст», «Минимальный возраст».
- Гистограмма «Распределение по возрастным группам» — смертность по сгруппированным возрастным категориям.
- Диаграмма «Распределение по полу».
- Гистограмма «Топ-10 диагнозов (МКБ-10)».

### 2.10.3. Разработка визуализации данных, превышающих городские и Республиканские показатели

Для построения круговых диаграмм на фронтенде были разработаны специализированные методы агрегации данных:

– **Детская смертность (stat_child_causes):** Реализован фильтр, который автоматически выбирает записи с кодами МКБ, начинающимися на «P» (Перинатальный период), «Q» (Врожденные аномалии), а также любые другие причины, если возраст умершего составляет менее 1 года. Это позволяет исключить ошибки ручного ввода и получить полную картину.

– **Материнская смертность (stat_maternal_causes):** Агрегация происходит по классу «O» (Беременность, роды и послеродовой период). Данные группируются по подклассам диагнозов с подсчетом процентного соотношения

Алматы қаласы цифрландыру басқармасы/Әндібай А.Э.

для выявления лидирующих причин патологий.

Внедрена система автоматического арбитража ответственности медицинской организации. На основе анализа кода МКБ-10 система маркирует случай:

– **Ответственность Стационара:** если причиной смерти стали прямые акушерские осложнения, тяжелые инфекции (сепсис), острые респираторные нарушения или врожденные пороки, требующие хирургического вмешательства.

– **Ответственность ПМСП (Поликлиники):** если причиной стали управляемые инфекции, хронические заболевания или состояния, которые должны были быть выявлены на этапе амбулаторного наблюдения.

Эта логика позволяет автоматически формировать KPI для главных врачей стационаров и поликлиник.

```
@action(detail=False, methods=['get'])
def stat_by_mkb_group(self, request):
    mkb_groups_config = {…

    raw_dataset = self.queryset.exclude(mkb__isnull=True).values(
        'mkb__mkb_code'
    ).annotate(
        m_count=Count('id')
    )

    def find_group_key(code_str):…

    others_count = 0
    for entry in raw_dataset:…

    result_list = []
    id_counter = 1

    for key, val in mkb_groups_config.items():…

    if others_count > 0:
        result_list.append({
            "id": id_counter,
            "range": "Other",
            "name": "Другие (S, T, и вне диапазонов)",
            "description": "Травмы, отравления и другие последствия воздействия внешних причин.",
            "count": others_count
        })

    result_list.sort(key=lambda x: x['count'], reverse=True)

    return Response(result_list)

@action(detail=False, methods=['get'])
def stat_by_death_cause(self, request):
```

Рисунок – 2.3.8.14. Фрагмент кода
алгоритма группировки по МКБ-10

Реализован функционал для анализа базы данных пациентов на Д-учете на основе анализа, полученного от аналитиков данных.

**Демографическое профилирование:**

– Реализован расчет возраста на момент постановки на учет и автоматическая сегментация по группам: 0-17, 18-44, 45-59, 60-74, 75+.

**Анализ динамики (YoY):**

– Разработан метод stats_dynamics_2021_2025, рассчитывающий тренды за 5 лет. Алгоритм вычисляет индекс роста относительно базового 2021 года (принят за 100%) и процент прироста к предыдущему году.

**KPI показатели:**

– Реализован быстрый расчет метрик: общее количество пациентов, средний возраст, гендерное распределение.

Алматы қаласы цифрландыру басқармасы\АДәндібай А. Ә.

## 2.11.Развитие модуля «Цифровая инклюзия»

**Цель**: Реализация дополнительного функционала для регистрации паспортов объектов и факта адаптации объектов.

**Задачи**:

4-й квартал:

- Разработка функционала по регистрации факта адаптации объекта.

**Заказчик**: Управление занятости и социальных программ города Алматы

### 2.11.1. Разработка функционала по регистрации факта адаптации объекта

В рамках выполнения задач по разработке функционала по регистрации факта адаптации объекта был разработан функционал, позволяющий фиксировать процесс адаптации каждого объекта.

Для объектов добавлена возможность регистрировать результаты инспекционных выездов, указывать даты проверок и вносить комментарии инспекторов.

Первый выезд фиксирует начальное состояние объекта: инспектор вносит дату визита, описание выявленных замечаний и выбирает результат («Исполнено», «Не исполнено», «Частично исполнено»). При отсутствии исполнения объект получает срок для устранения нарушений.

Рисунок – 2.3.9.1. Инспекция по адаптации объекта, первый этап

Также разработан функционал, предоставляющий возможность проведения повторной проверки.

Для регистрации факта адаптации объекта, инспектор фиксирует дату повторного выезда и фиксирует в системе результаты повторного выезда на объект. В случае если на объекте снова не были выполнены требования, система автоматически предусматривает передачу информации в уполномоченный орган.

Уполномоченный орган осуществляет финальную проверку: фиксирует дату инспекции, описывает оставшиеся нарушения или устранённые замечания, и принимает меры наказания — от предупреждения до административных санкций.

Если объект соответствует требованиям, результат проверки устанавливается как «Соответствует».

Алматы қаласы цифрландыру басқармасы\Адамбай А. Ә.

Рисунок – 2.3.9.2. Инспекция по адаптации объекта, второй этап

Рисунок – 2.3.9.3. Проверка результатов уполномоченного органа

### 2.12. Развитие модуля «Аналитика Open Almaty»

**Цель**: Создать интерактивной панели для руководства города Алматы по мониторингу обращений жителей.

**Задачи**:

4-й квартал:

- Разработка рейтинга Исполнительных органов с помощью LLM обработки обращений, ответов, комментариев
- Реализация модуля по согласованному макету

**Источник данных**: АПК «Open Almaty», Телеграмм канал «Таза Казахстан»

**Заказчик**: Пресс служба акимата города Алматы

#### 2.12.1. Разработка рейтинга Исполнительных органов с помощью LLM обработки обращений, ответов, комментариев

Целью обработки сообщений с помощью LLM является разработка прозрачного и воспроизводимого рейтинга исполнительных органов, основанного не на формальных показателях (количество закрытых обращений, доля ответов в срок), а на качестве работы с обращениями граждан.

Для реализации данного механизма применяется следующая связка:

- операционных данных (сроки, каналы, статус обращения);
- обратной связи самих заявителей (словесные оценки «решено», «частично», «не решено» и т.д.);
- оценки качества ответов с помощью LLM-модели, которая автоматически анализирует текст ответа и комментарий заявителя.

Алматы қаласының цифрлық даму басқармасы АДәндібай А. Ә.

Результатом является финальный балл по каждому обращению, который последующем агрегируется в рейтинги акиматов, управлений и других исполнительных органов.

***Источники данных и каналы обращений***

В расчёте учитываются обращения граждан, поступающие через различные каналы:

- Портал и фронт-офис OpenAlmaty;
- колл-центр;
- социальные сети и мессенджеры, интегрированные в систему.

Для каждого обращения в системе доступны:

- дата и время поступления обращения и дата/время первого ответа;
- исполнительный орган и конкретный исполнитель;
- текст обращения;
- текст ответа;
- при наличии — комментарий/оценка заявителя после получения ответа;
- дополнительная служебная информация (статус, сроки, НПА, прикреплённые файлы и др.).

***Методика оценки качества ответов***

Оценка по каждому обращению строится на четырёх ключевых компонентах:

- Скорость ответа;
- Качество по словам (словесная оценка заявителя);
- Оценка ИИ (LLM) — анализ качества ответа и комментария;
- Финальный интегральный балл (FinalScore).

Скорость фиксирует – сроки обработки обращений исполнителем относительно нормативов:

- для Портала и фронт-офиса норматив — 15 рабочих дней;
- для колл-центра и соцсетей — 3 дня.

Если ответ был дан в пределах норматива, скорость считается максимальной (Speed = 1.0).

Чем больше задержка относительно норматива, тем ниже значение показателя скорости (по непрерывной шкале от 0 до 1).

Таким образом, скорость одновременно отражает:

- соблюдение установленных регламентов;
- фактическую оперативность реагирования на запросы граждан.

Если гражданин после получения ответа оставляет текстовую\голосовую оценку, она переводится в числовую по шкале от 0…1:

- формулировки типа «решено» или «отлично» → значения, близкие к 1.0;
- оценки «частично» → средний уровень;

Алматы қаласы цифрландыру басқармасы\АДендібай А. Ә.

– «не решено», негативные комментарии или жалобы → значения, близкие к 0.

Таким образом, в расчёт напрямую включается субъективное восприятие гражданина, но переведённое в формализованный числовой формат.

Для автоматизированного анализа качества ответов исполнителей и комментариев заявителей в проекте используется не абстрактная LLM-модель, а конкретный сервис Mistral API.

Это современная large language model, к которой обращение осуществляется через REST-интерфейс из внутреннего аналитического контура.

***Архитектура вызова***

1) Из операционной базы для каждого обращения формируется структурированный запрос к Mistral API, включающий:
   – текст обращения гражданина;
   – текст официального ответа;
   – (при наличии) комментарий заявителя после ответа;
   – служебные признаки (канал поступления, тип обращения и т.п., без персональных данных).
2) Mistral возвращает машиночитаемый ответ в заранее зафиксированном формате (JSON), который содержит:
   – числовую оценку качества ответа исполнителя в шкале 0…1;
   – числовую оценку комментария заявителя (удовлетворён/не удовлетворён);
   – вспомогательные флаги и текстовые пояснения (например, есть ли адрес, сроки, конкретные меры, ссылка на НПА).

Логика оценки, реализованная через технологию Mistral. При обработке каждого кейса модель Mistral анализирует:

– структурированность ответа: указан ли конкретный адрес/объект, есть ли чётко прописанные сроки и ответственный орган;
– юридическую корректность: присутствует ли ссылка на нормативно-правовые акты, регламенты, программы;
– фактическое решение вопроса: описаны ли реальные действия/меры, а не только формальные отписки;
– тон и содержание комментария заявителя: присутствует ли благодарность, подтверждение решения проблемы либо, наоборот, указание на формальность и повторяющиеся жалобы.

Все эти аспекты объединяются в две числовые метрики:

– оценка качества ответа исполнителя (0…1);
– оценка реакции заявителя (0…1).

Далее все показатели входят в расчёт FinalScore с заданными весами (для Портала — 50% ответ + 10% комментарий + 20% словесная оценка + 20% скорость; для других каналов — 70% ответ + 30% скорость).

***Расчёт финального балла по обращениям. Обращения через Портал и***

Алматы қаласы цифрлаңыру басқармасы\АДәндібай А. О.

***фронт-офис***

Для обращений, поступивших через Портал и фронт-офис, финальный балл рассчитывается как взвешенная комбинация:

- 50% — качество текста ответа (по оценке ИИ);
- 10% — анализ комментария заявителя (по оценке ИИ);
- 20% — словесная оценка заявителя («решено», «отлично», «частично», «не решено»);
- 20% — скорость ответа.

Таким образом, наиболее значимым компонентом остаётся содержание ответа, при этом учитываются и мнение гражданина, и соблюдение сроков.

***Обращения из других каналов***

Для остальных каналов (например, колл-центр, соцсети), где не всегда доступна полная обратная связь заявителя, применяется упрощённая формула:

- 70% — качество ответа (по оценке ИИ);
- 30% — скорость ответа.

Это позволяет сохранять сопоставимость баллов по разным каналам, при этом не завышая оценку там, где отсутствует явная оценка гражданина.

***Интерпретация шкалы FinalScore***

Финальный балл FinalScore находится в диапазоне от 0 до 1 и интерпретируется следующим образом:

0.00 – 0.39 — слабый ответ, высокая формальность, отсутствие конкретики и/или значительная задержка;

0.40 – 0.59 — результат ниже среднего: ответ формально дан, но недостаточно конкретен либо не обеспечивает решение проблемы;

0.60 – 0.79 — приемлемый уровень: в целом по делу, с достаточным количеством информации;

0.80 – 1.00 — хороший и полный ответ, понятный гражданину, с конкретными сроками и ответственными.

***Формирование рейтинга исполнительных органов***

После расчёта FinalScore для каждого обращения выполняется агрегирование на уровне органов:

1) Для каждого исполнительного органа рассчитываются:
   - средний FinalScore по всем обращениям за период;
   - медианный балл;
   - доля обращений с низкой оценкой (например, FinalScore < 0.4);
   - доля обращений с высокой оценкой (FinalScore ≥ 0.8);
   - распределение по каналам (портал, колл-центр, соцсети).
2) Для корректности сравнения вводятся минимальные пороги по количеству обращений (например, орган учитывается в рейтинге, если за период было получено не менее N обращений).
3) На основе средневзвешенного FinalScore формируется сводный рейтинг исполнительных органов, который может отображаться:

Алматы қаласы цифрландыру басқармасы\АДәндібай А. Ә.

– в виде ранжированного списка;
– в виде цветовой шкалы по картосхеме города;
– по категориям (районные акиматы, управления, службы поддержки).

***Роль LLM-модели и преимущества подхода***

Использование LLM-модели в данном контуре обеспечивает:

– масштабируемость — возможность анализировать десятки тысяч ответов без ручной выборки;
– единообразие критериев — все ответы оцениваются по одним и тем же правилам, без субъективного влияния конкретного сотрудника;
– глубину анализа — модель способна отличить формальные отписки от содержательных ответов с указанием адреса, сроков и НПА;
– связку содержания с восприятием гражданина — одновременный учёт комментариев заявителя и текста ответа.

В итоге рейтинг исполнительных органов основан не только на «галочке о закрытии обращения», но и на реальном качестве коммуникации с жителями, что позволяет:

– выявлять сильные и слабые стороны конкретных служб;
– адресно выстраивать обучение и изменение регламентов;
– использовать рейтинг как инструмент управленческой мотивации.

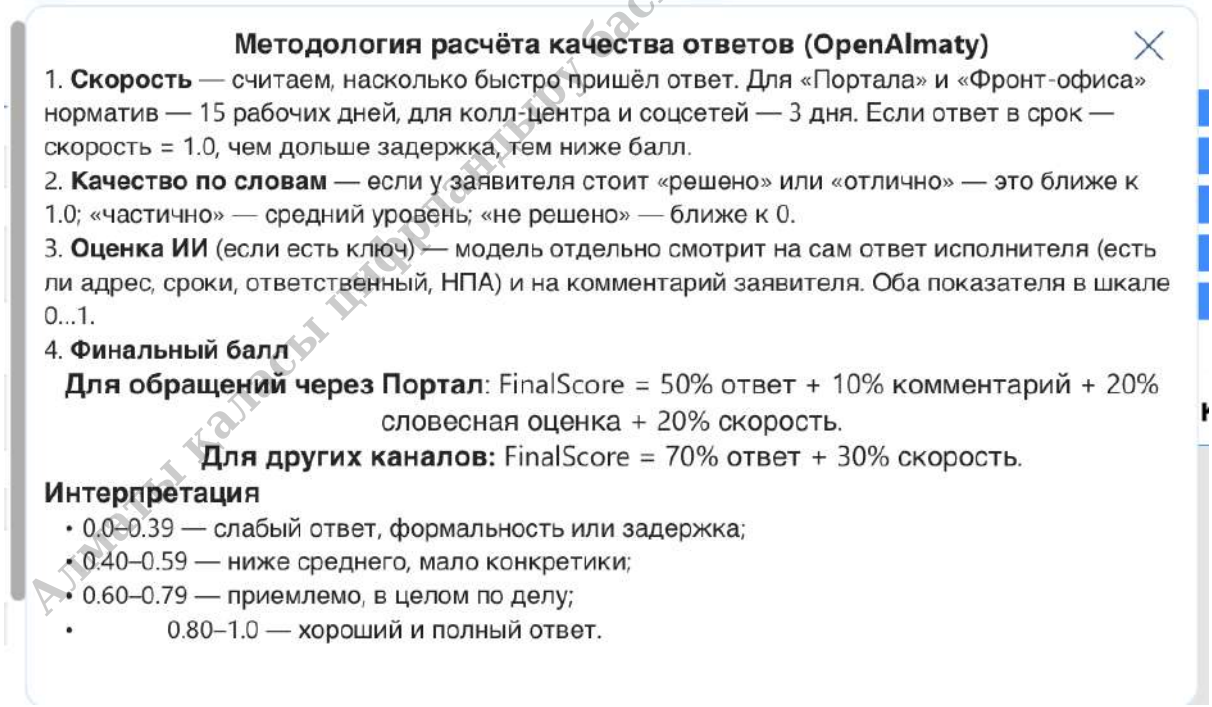



Рисунок – 2.3.10.1. Методология расчета качества ответов

### 2.12.2. Реализация модуля по согласованному макету

На основании использования LLM-модели был разработан дашборд, который отображает результаты.

Алматы қаласы цифрлық басқармасы\Ландбой А. Э.

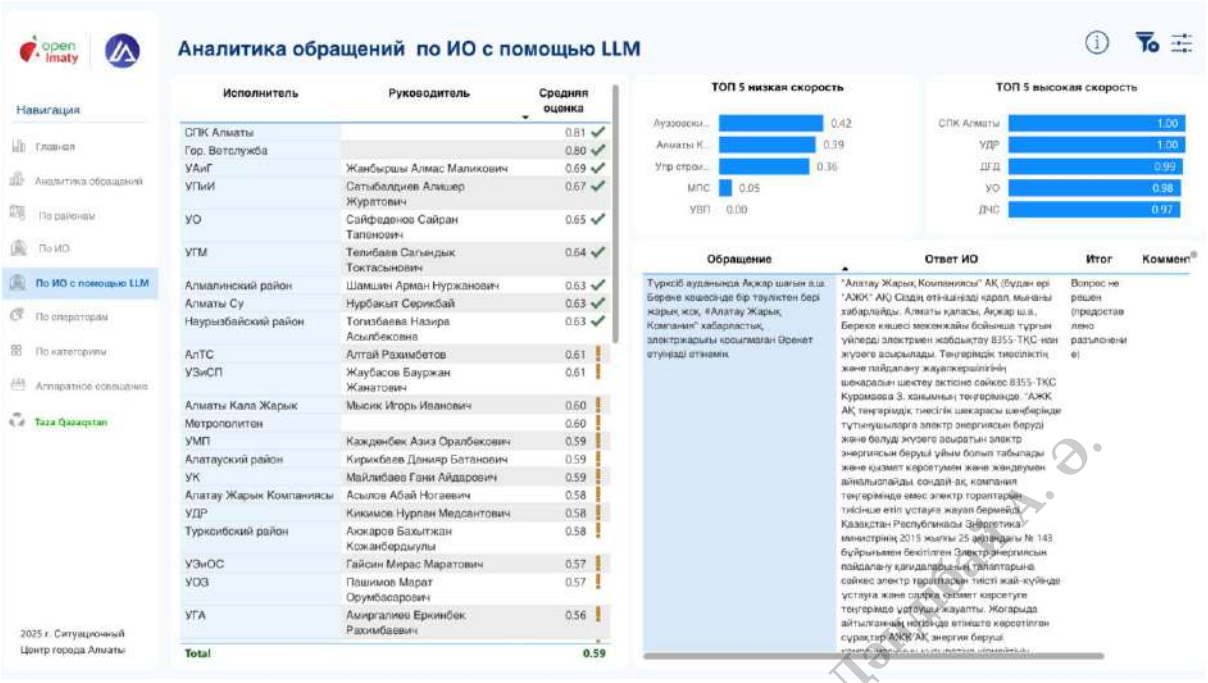



Рисунок – 2.3.10.2. Аналитика обращений по ИО с помощью LLM

Дашборд отображает оценку качества ответов исполнителей (ИО) на обращения граждан.

В дашборде представлены следующие компоненты:

1) Таблица исполнителей, которая состоит из следующих параметров:
   - Перечень ИО;
   - Руководители;
   - Средняя оценка качества ответа, рассчитанная с помощью LLM;
   - Индикаторы (галочки/восклицательные знаки), показывающие уровень качества.

Таблица исполнителей показывает оценку качества работы каждой организации-исполнителя по обращениям граждан.

2) Рейтинги скорости обработки отображают следующее:
   - ТОП-5 с самой низкой скоростью ответов;
   - ТОП-5 с самой высокой скоростью.

Рейтинги скорости обработки отображают насколько быстро разные исполнители отвечают на обращения граждан.

3) Карточка обращения:
   - Текст обращения.
   - Ответ организации-исполнителя.
   - Итог (например: проблема решена / не решена, предоставлено разъяснение и т.д.).
   - Возможные комментарии.

Карточка обращения отображает подробную информацию по конкретному обращению гражданина и ответу исполнителя.

Алматы қаласы цифрландыру басқармасы(Дәстүрбаев А. Ә.

### 2.12.3. Государственная программа «Taza Qazaqstan»

«Taza Qazaqstan» — это государственная программа, направленная на улучшение качества городской среды, санитарного состояния, озеленения и общественной инфраструктуры. Одним из ключевых инструментов реализации инициативы является цифровая система обработки обращений граждан, интегрированная в единую платформу Open Almaty и дополненная собственными аналитическими контурами.

Цель проекта — обеспечить:

- оперативное реагирование городских служб на обращения населения;
- прозрачность и подотчётность исполнительных органов;
- анализ проблем городской среды на основе данных;
- приоритизацию мероприятий по благоустройству, уборке, ремонту и санитарному содержанию.

***Структура данных и информационная панель***

На основании схемы, которая была приложена, система «Taza Qazaqstan» опирается на модульно организованную базу данных.

Основные сущности:

1) *tazaqazaq_request* — обращения граждан

Содержит ключевую информацию:

- адрес и геопозицию;
- категорию обращения;
- дату создания;
- связанный регион;
- источник (платформа);
- ID пользователя, подавшего заявку.

Каждое обращение имеет чёткую привязку к территории, что позволяет строить детальную пространственную аналитику.

2) *tazaqazaq_region* — административные и функциональные зоны

Определяют:

- районирование города;
- популярность или приоритет зоны;
- иерархию регионов (главный/дочерний регион).

Используются для группировки обращений по районам, микрорайонам, кварталам.

3) *tazaqazaq_request_category* — категории обращений

Позволяют классифицировать заявки по типам:

- санитарная очистка;
- вывоз мусора;
- засорённые территории;
- проблемы озеленения;

Алматы қаласы цифрландыру басқармасы\АДәндібай А.Ә.

– несанкционированные свалки;
– содержание улиц, тротуаров, парков и т.д.

Каждая категория имеет:

– нормативные сроки исполнения (execution_days),
– приоритет;
– названия на трёх языках.

4) *tazaqazaq_organization* — ответственные службы

Хранят данные об организациях и ведомствах:

– управления;
– районные акиматы;
– коммунальные службы;
– подрядчики.

5) *tazaqazaq_department* — исполнители конкретного уровня

Для каждого обращения определяется ответственный департамент внутри организации.

***Логика обработки обращений***

Когда гражданин подаёт заявку (через портал Open Almaty, мобильное приложение или фронт-офис), система:

– сохраняет обращение в таблицу *tazaqazaq_request*;
– определяет регион, категорию и ответственный орган;
– запускает нормативный таймер исполнения;
– отображает заявку на карте для мониторинга;
– передаёт исполнителям в рабочие системы для выполнения;
– фиксирует результат и комментарий заявителя.

***Аналитический дашборд (Power BI / Open Almaty Analytics)***

Аналитическая панель строится вокруг следующих ключевых метрик:

1) Количество обращений

Общее число обращений за период — на дашборде пример: 11,671.

2) Текущая карта обращений

Географическое распределение обращений визуализируется в виде тепловых пятен/точек. Это позволяет:

– выявлять «горячие зоны» загрязнения;
– контролировать работу подрядчиков и служб в каждом районе;
– отслеживать накопление проблем в определённых локациях.

3) Фильтры и разрезы

Аналитика доступна по:

– категориям обращений;
– районам города;
– организациям и подразделениям;
– статусам исполнения (в срок, просрочено, закрыто).

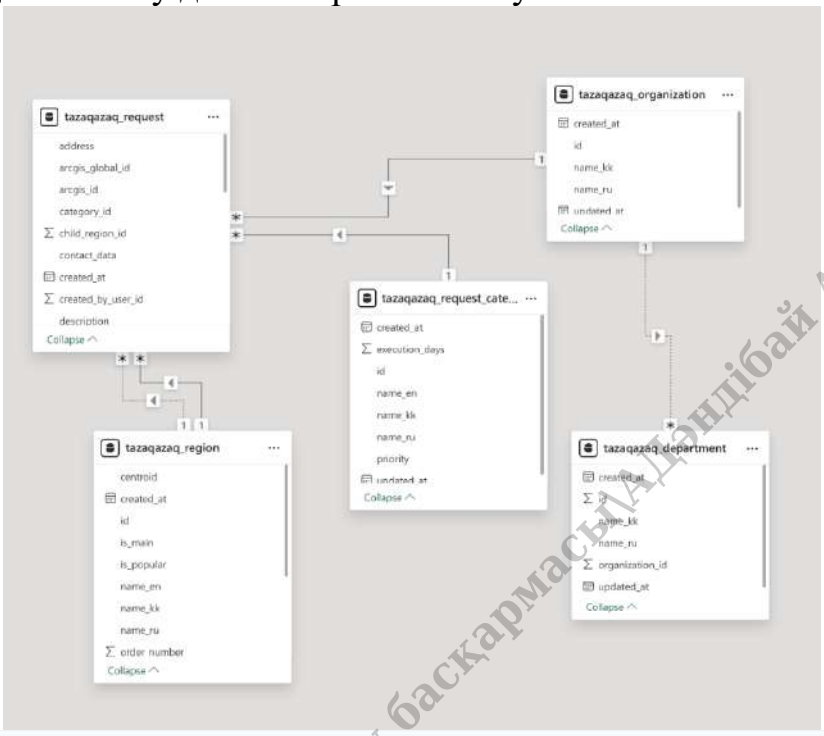
4) Выполнение сроков

Алматы қаласы цифрландыру басқармасы\АДәнгібай А. Ә.

Дашборд отображает:

- количество обращений в срок;
- просроченные выполнения;
- динамику исполнения по районам.

Это создаёт основу для KPI органов и служб.



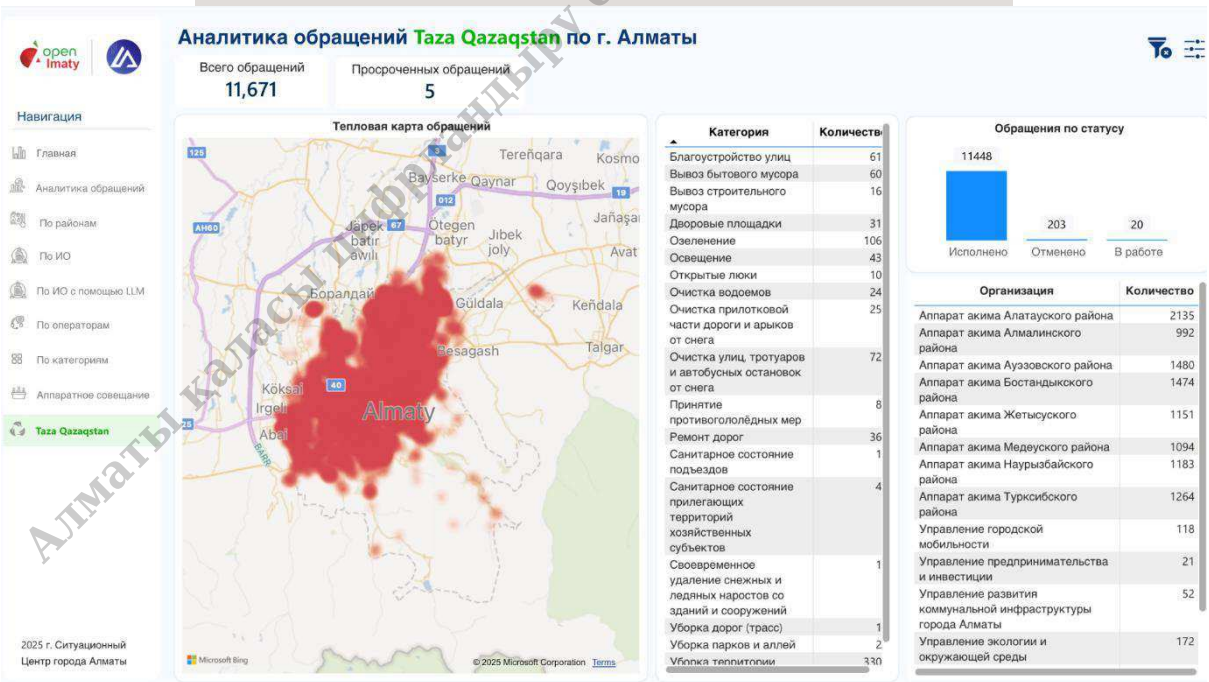




Рисунок – 2.3.10.3. Аналитика обращений «Taza Qazaqstan»

Алматы қаласы цифрландыру басқармасы Әдәнбай А. Ә.

## 2.13. Развитие модуля «Аналитика нагрузки дорог»

***Цель:***

Развить функционал модуля для учета новых данных и улучшения предикативной аналитики по износу дорожного полотна.

***Задачи:***

4-й квартал:

- Формирование индекса состояния дороги (PCI), классов и рекомендаций.
- Разработка усовершенствованных алгоритмов предикативной аналитики износа дорог.
- Создание интерфейса для визуализации зон повышенного износа на основе новых данных.

***Заказчик:*** Проектный офис ГРБН «Развитие транспортной системы», Управление городской мобильности

### 2.13.1. Формирование индекса состояния дороги (PCI), классов и рекомендаций

Для автоматизированной оценки качества дорожного покрытия был реализован модуль расчёта индекса состояния дороги (PCI) на основе анализа фотоматериалов. Данная подсистема обеспечивает единообразную классификацию участков дорожной сети, формирование рекомендаций по ремонту и приоритизацию мероприятий по содержанию дорог.

**Цель разработки:** создать механизм, позволяющий автоматически определять текущее состояние дорожного покрытия по фотоснимкам, присваивать ему интегральный индекс PCI (0–100), относить участок к соответствующему классу состояния и формировать рекомендации по необходимым видам ремонтных работ и их срочности. Это обеспечивает объективность, масштабируемость и оперативность мониторинга состояния дорожной инфраструктуры.

Описание алгоритма:

- Первичный анализ фотоснимков. Фотоматериалы дороги обрабатываются моделью компьютерного зрения Mistral (Pixtral), которая распознаёт ключевые типы повреждений покрытия: ямы, трещины, выкрашивания, колейность, оголение щебня, а также оценивает уровень риска для безопасности движения.

Модель формирует предварительную оценку общего состояния участка и прогнозируемый индекс PCI. Все результаты приводятся к единому формату для дальнейшей обработки.

- Определение класса состояния покрытия. На основе совокупности данных о характере и интенсивности повреждений и уровне риска система присваивает каждому участку один из пяти классов: excellent, good, fair, poor, critical. Это позволяет согласовать фактические признаки дефектов с итоговым состоянием и минимизировать погрешности автоматического анализа.

Алматы қаласы цифрландыру басқармасы\АДәндібай А. Ә.

• Корректировка числового индекса PCI. Предварительный показатель PCI приводится в соответствие с диапазоном, установленным для конкретного класса состояния, что обеспечивает точность и сопоставимость индексов между различными участками.

• Формирование эксплуатационных рекомендаций. Для каждого участка автоматически определяются:

— характер необходимых работ (от наблюдения и локальной герметизации трещин до капитального ремонта);

— приоритет выполнения работ;

— степень риска для автомобилистов;

— необходимость ограничения скорости;

— индекс тяжести повреждений, отражающий глубину деградации покрытия.

• Систематизация данных. Фотоматериалы сортируются по папкам в соответствии с присвоенными классами состояния, что облегчает визуальный контроль и последующую аналитическую обработку.

• Формирование аналитической базы. В единую базу включаются: PCI, категория состояния, рекомендации по ремонту, уровень риска, сводка дефектов, приоритет работ, тип предполагаемого вмешательства и дополнительные характеристики участка дороги.

Алматы қаласының ... пайдаланылған ... Данибай А. Ә.

```python
def encode_image(path: Path) -> str:
    with open(path, "rb") as f:
        return base64.b64encode(f.read()).decode("utf-8")


_key_index = 0
_client_cache = {}
_client_lock = threading.Lock()


def get_client():
    """Round-robin по ключам, кэшируем клиентов (потокобезопасно)."""
    global _key_index
    with _client_lock:
        key = API_KEYS[_key_index % len(API_KEYS)]
        _key_index += 1

        if key not in _client_cache:
            _client_cache[key] = Mistral(api_key=key)

        return _client_cache[key], key


def extract_json(raw: str):
    raw = raw.strip()
    try:
        return json.loads(raw)
    except Exception:
        pass

    s = raw.find("{")
    e = raw.rfind("}")
    if s != -1 and e != -1 and e > s:
        snippet = raw[s:e+1]
        try:
            return json.loads(snippet)
        except Exception:
            return None
    return None


def compute_condition(data):
    poth = str(data.get("potholes", "none")).lower()
    cracks = str(data.get("cracks", "none")).lower()
    rav = str(data.get("raveling", "none")).lower()
    rut = str(data.get("rutting", "none")).lower()
    exposed = bool(data.get("exposed_aggregate", False))
    risk = str(data.get("safety_risk", "low")).lower()

    # ЛОГИКА НЕ МЕНЯЛ
    if poth in ("many", "severe") or risk == "high":
        if exposed or rav == "severe" or cracks == "severe":
            return "critical"
        return "poor"
```

Рисунок – 2.3.11.1. Фрагмент кода

```
SYSTEM_PROMPT = """
Ты — эксперт по оценке состояния автомобильных дорог по фотографиям.

1. СЧИТАЕМ РЕАЛЬНЫМИ ДЕФЕКТАМИ:
   - ямы, выбоины, провалы
   - разрушение покрытия
   - рваные трещины с раскрытием
   - сеточные трещины (если это не тень)
   - оголенный щебень, выкрашивание
   - сильная колейность

2. НЕ СЧИТАЕМ ДЕФЕКТАМИ (игнорируй):
   - тени от деревьев, машин, ограждений
   - мокрый асфальт, влажные пятна
   - грязь, следы шин
   - ровные швы между слоями/плитами
   - тонкие волосные трещины без разрушения

3. УРОВНИ СОСТОЯНИЯ (overall_condition):
- "excellent" — почти нет дефектов, покрытие ровное.
- "good" — немного мелких трещин, без ям и серьёзных дефектов.
- "fair" — много трещин/усталое покрытие, но без крупных ям.
- "poor" — разрушения, отдельные ямы, заметное ухудшение.
- "critical" — крупные ямы, провалы, сильное разрушение покрытия.

PCI (0-100) привязывай к этим диапазонам:
- excellent: 90-100
- good: 80-89
- fair: 50-79
- poor: 20-49
- critical: 0-19

Отвечай строго JSON без лишнего текста.
"""

USER_PROMPT = """
Проанализируй фото дороги и верни JSON строго такого вида:

{
  "potholes": "none | few | many | severe",
  "cracks": "none | hairline | moderate | severe",
  "raveling": "none | moderate | severe",
  "rutting": "none | moderate | severe",
  "exposed_aggregate": true/false,
  "safety_risk": "low | medium | high",
  "overall_condition": "excellent | good | fair | poor | critical",
  "model_pci_score": 0-100,
  "explanation_short": "кратко, до 40 слов, на русском"
}
"""
```

Рисунок – 2.3.11.2. Фрагмент кода

```
def process_image(img: Path):
    data = analyze_image(img)

    if not data:
        cond = "fair"
        pci = 50
        risk = "medium"
        expl = "Ошибка анализа модели"
        defects_summary = "Нет данных (ошибка анализа модели)."
    else:
        final_cond = compute_condition(data)
        pci = adjust_pci(final_cond, data.get("model_pci_score", 60))
        cond = final_cond
        risk = str(data.get("safety_risk", "low")).lower()
        expl = str(data.get("explanation_short", "")).strip()
        defects_summary = build_defects_summary(data)

    recommendation = get_recommendation(cond, risk)
    risk_detail = build_risk_detail(cond, risk)
    maintenance_priority = get_maintenance_priority(cond, risk)
    recommended_action_type = get_recommended_action_type(cond, risk)
    needs_speed_limit = get_needs_speed_limit(cond, risk)
    severity_index = compute_severity_index(pci)
    image_side = get_image_side(img.name)

    # Пока lighting / weather не просим у модели отдельно, оставляем пустыми
    lighting = ""
    weather = ""

    out_dir = OUTPUT_BASE / PACKS[cond]
    shutil.copy2(img, out_dir / img.name)

    logging.info(f"{img.name} → {cond.upper()} | PCI {pci} | RISK {risk}")

    return {
        "file": img.name,
        "condition": cond,
        "pci": pci,
        "risk": risk,
        "risk_detail": risk_detail,
        "recommendation": recommendation,
        "defects_summary": defects_summary,
        "maintenance_priority": maintenance_priority,
        "recommended_action_type": recommended_action_type,
        "needs_speed_limit": needs_speed_limit,
        "severity_index": severity_index,
        "image_side": image_side,
        "lighting": lighting,
        "weather": weather,
        "explanation": expl,
    }
```

Рисунок – 2.3.11.3. Фрагмент кода

Алматы қаласы ... қарма ... ий А. Ә.

### 2.13.2. Разработка усовершенствованных алгоритмов предикативной аналитики износа дорог

Данная методика представляет собой комплексный подход к оценке транспортной нагрузки на дорожные сегменты и прогнозированию деградации дорожного покрытия.

Источником данных о транспортном потоке служит система камер Cepgek, фиксирующая тип транспортных средств, направление движения, интенсивность потока и сопутствующие параметры. В сочетании с инженерной оценкой состояния дорожного полотна — индексом PCI — методика позволяет строить детальные прогнозы износа, определять ожидаемый срок службы покрытия и формировать приоритеты по ремонтам.

Дорога изнашивается неравномерно: разные сегменты испытывают различную нагрузку в зависимости от структуры трафика, доли грузовых автомобилей, интенсивности потоков в разных направлениях, сезонных колебаний и других факторов. Современные системы мониторинга позволяют получать высокочастотные данные о движении транспорта, а индекс PCI обеспечивает объективную оценку текущего состояния покрытия.

**Цель методики:** объединить данные о транспортной нагрузке и состоянии покрытия в единую математическую модель, которая позволяет:

- количественно оценивать нагрузку на каждом сегменте дороги;
- учитывать различия между потоками в разных направлениях;
- моделировать изменение состояния покрытия во времени;
- прогнозировать момент достижения критического состояния дороги;
- определять приоритетность ремонта на основе фактических и измеримых параметров.

Каждая камера Cepgek установлена рядом с определённым участком дороги. Для корректного сопоставления её данных с центролиниями дорожных сегментов используется следующий подход:

Определяется окрестность камеры (радиус 20–30 м) и выбирается наиболее вероятный сегмент дороги.

Так как камеры фиксируют движение в двух направлениях, сопоставляется направление фиксации с осевой линией дороги.

Поток в направлении **dir1** относится к сегментам, ориентированным из точки А в точку Б, а поток **dir2** — из точки Б в точку А.

***Для каждого типа ТС*** и для каждого направления движения определяется среднегодовое количество проходов:

$$E_{\text{year}}^{(d)} = \sum_i N_i^{(d)} \cdot \gamma_i$$

Алматы, қаласы цифрландыру бастармасы|Адондбай А. Б.

- $E_{\text{year}}^{(d)}$ — эквивалентная нагрузка в направлении $d$
- $N_i^{(d)}$ — годовое число проездов ТС типа $i$
- $\gamma_i$ — коэффициент нагрузки для ТС типа $i$

Каждое транспортное средство «взвешивается» по своей нагрузке.

***Общая годовая нагрузка на сегмент дороги***

$$E_{\text{year}} = E_{\text{year}}^{(\text{dir1})} + E_{\text{year}}^{(\text{dir2})}$$

Учитываются оба направления движения. Для расчёта итоговой нагрузки, которая влияет на скорость износа покрытия.

***Связь нагрузки и ежегодного снижения PCI***

$$\Delta\text{PCI}_{\text{year}} = k \cdot E_{\text{year}}$$

- $\Delta\text{PCI}_{\text{year}}$ — на сколько падает PCI за 1 год
- $k$ — коэффициент чувствительности покрытия (калибруется по историческим данным)
- $E_{\text{year}}$ — годовая нагрузка (рассчитана выше)

Если транспортная нагрузка высока → PCI падает быстрее.

***Прогноз PCI на будущие годы***

$$\text{PCI}(t+1) = \text{PCI}(t) - k \cdot E_{\text{year}}$$

PCI в следующем году = PCI в текущем году − ежегодное снижение

Формула линейная, применима при стабильном трафик

Если нагрузка по типам ТС каждый код меняется:

$$\text{PCI}(t+1) = \text{PCI}(t) - k \cdot E_{\text{year}}(t+1)$$

**5.** ***Определение срока службы покрытия***

PCI через n лет:

$$\text{PCI}(t_{\text{now}} + n) = \text{PCI}_{\text{now}} - n \cdot k \cdot E_{\text{year}}$$

Условие достижения критического уровня

$$\text{PCI}_{\text{crit}} = \text{PCI}_{\text{now}} - n \cdot k \cdot E_{\text{year}}$$

Решение: сколько лет прослужит дорога:

$$n = \frac{\text{PCI}_{\text{now}} - \text{PCI}_{\text{crit}}}{k \cdot E_{\text{year}}}$$

Год отказа покрытия:

Алматы қаласы цифрландыру басқарласы\Дәндібай А. Ә

$$t_{\text{fail}} = t_{\text{now}} + n$$

- $\text{PCI}_{\text{now}}$ — текущий PCI
- $\text{PCI}_{\text{crit}}$ — критический уровень
- $t_{\text{now}}$ — текущий год

```python
def compute_pci_drop_and_lifetime(E_year: float,
                                  pci_now: float = PCI_NOW,
                                  pci_crit: float = PCI_CRIT,
                                  k: float = K_SENSITIVITY,
                                  t_now: int = T_NOW) -> dict:
    """
    Считает:
      - ежегодное падение PCI: dPCI_year = k * E_year
      - сколько лет до достижения критического PCI
      - год t_fail, когда PCI достигнет критического уровня.
    """
    if E_year <= 0 or k <= 0:
        return {
            "pci_drop_year": 0.0,
            "years_to_crit": np.inf,
            "t_fail": np.nan
        }

    d_pci_year = k * E_year  # ΔPCI_year = k * E_year

    if d_pci_year <= 0:
        return {
            "pci_drop_year": 0.0,
            "years_to_crit": np.inf,
            "t_fail": np.nan
        }

    # n = (PCI_now - PCI_crit) / (k * E_year)
    years_to_crit = (pci_now - pci_crit) / d_pci_year

    if years_to_crit < 0:
        # Уже хуже критического
        t_fail = t_now
        years_to_crit = 0.0
    else:
        t_fail = t_now + years_to_crit

    return {
        "pci_drop_year": d_pci_year,
        "years_to_crit": years_to_crit,
        "t_fail": t_fail
    }
```

Рисунок – 2.3.11.4. Фрагмент кода

Алматы ка... ...ий А. Э.

```python
def compute_equivalent_load(row) -> float:
    """
    Считает эквивалентную годовую нагрузку на основе колонок с типами ТС.
    E_year = sum_i N_i * gamma_i
    """
    total = 0.0
    for vt in VEHICLE_TYPES:
        n_i = row.get(vt, 0)
        gamma_i = GAMMA.get(vt, 1.0)
        total += n_i * gamma_i
    return total
```

Рисунок – 2.3.11.5. Фрагмент кода

### 2.13.3. Создание интерфейса для визуализации зон повышенного износа на основе новых данных

Проведены работы по улучшению производительности и функциональности сервиса геоданных.

Модифицирован сервис генерации векторных тайлов (MVT Service). Улучшены алгоритмы запросов к PostGIS, что привело к значительному ускорению отрисовки слоев на карте, особенно при работе с большими объемами данных.

```python
def get_mvt_tile(self, z: int, x: int, y: int, gri_class=None, district=None, density_level=None):
    max_val = 2**z - 1
    if x < 0 or y < 0 or x > max_val or y > max_val:
        return None

    filters = []
    params = [z, x, y]

    if gri_class:
        op = ' = ANY(%s)' if isinstance(gri_class, (list, tuple)) else ' = %s'
        filters.append(f"t."GRI_class"{op}")
        params.append(gri_class)

    # Логика для district
    if district:
        op = ' = ANY(%s)' if isinstance(district, (list, tuple)) else ' = %s'
        filters.append(f"t.district{op}")
        params.append(district)

    # --- НОВЫЙ БЛОК ДЛЯ DENSITY_LEVEL ---
    if density_level:
        op = ' = ANY(%s)' if isinstance(density_level, (list, tuple)) else ' = %s'
        filters.append(f"t.density_level{op}")
        params.append(density_level)

    where_clause = "AND " + " AND ".join(filters) if filters else ""

    sql = f"""
        WITH bounds AS (
            SELECT ST_Transform(ST_TileEnvelope(%s, %s, %s), 4326) AS geom
        )
        SELECT ST_AsMVT(tile, 'geo_risk', 4096, 'geom') AS mvt
```

Алматы ... шифрландыру баскармасы\АЛәндібай А...

services > pci.service.ts > ...

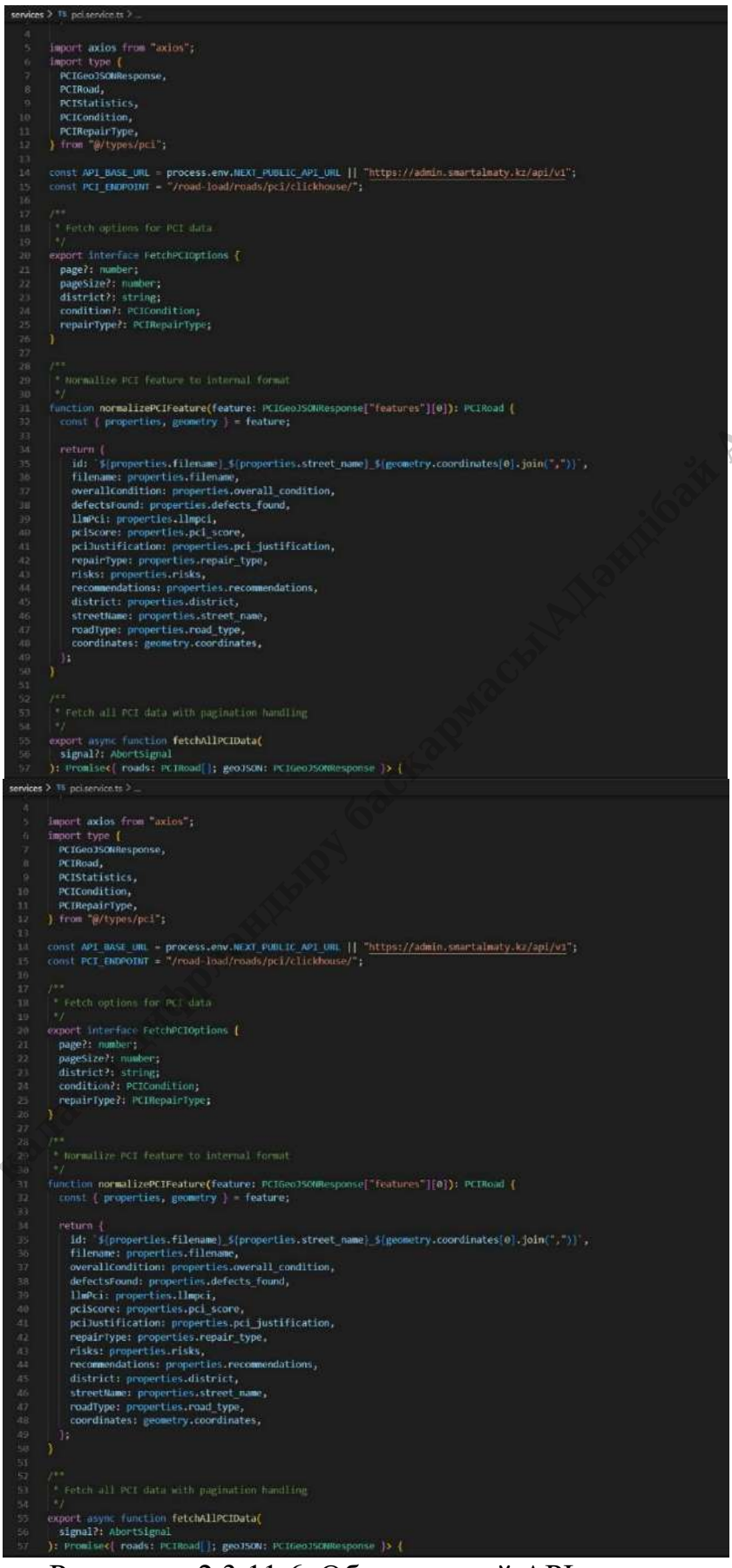
C:\Users\User\Desktop\road_4kv\roads\services • Contains emphasized items

```
    const firstResponse = await axios.get<PCIGeoJSONResponse>(
      }
    );

    totalCount = firstResponse.data.count;
    allFeatures.push(...firstResponse.data.features);
    nextUrl = firstResponse.data.next;

    // Fetch remaining pages if needed
    while (nextUrl && !signal?.aborted) {
      // Fix protocol to HTTPS (API returns http:// which violates CSP)
      const secureUrl = nextUrl.replace(/^http:\/\//, "https://");

      const response = await axios.get<PCIGeoJSONResponse>(secureUrl, {
        signal,
      });

      allFeatures.push(...response.data.features);
      nextUrl = response.data.next;
    }

    // Normalize features to internal format
    const roads = allFeatures.map(normalizePCIFeature);

    // Create GeoJSON for map display
    const geoJSON: PCIGeoJSONResponse = {
      type: "FeatureCollection",
      count: totalCount,
      next: null,
      previous: null,
      features: allFeatures,
    };

    return { roads, geoJSON };
  } catch (error) {
    if (signal?.aborted) {
      throw new Error("Request aborted");
    }
    console.error("Error fetching PCI data:", error);
    throw error;
  }
}

/**
 * Fetch PCI data with filters (single page)
 */
export async function fetchPCIData(
  options: FetchPCIOptions = {},
  signal?: AbortSignal
): Promise<PCIGeoJSONResponse> {
  try {
    const params: Record<string, any> = {};
```

services > pci.service.ts > ...

```
import axios from "axios";
import type {
  PCIGeoJSONResponse,
  PCIRoad,
  PCIStatistics,
  PCICondition,
  PCIRepairType,
} from "@/types/pci";

const API_BASE_URL = process.env.NEXT_PUBLIC_API_URL || "https://admin.smartalmaty.kz/api/v1";
const PCI_ENDPOINT = "/road-load/roads/pci/clickhouse/";

/**
 * Fetch options for PCI data
 */
export interface FetchPCIOptions {
  page?: number;
  pageSize?: number;
  district?: string;
  condition?: PCICondition;
  repairType?: PCIRepairType;
}

/**
 * Normalize PCI feature to internal format
 */
function normalizePCIFeature(feature: PCIGeoJSONResponse["features"][0]): PCIRoad {
  const { properties, geometry } = feature;

  return {
    id: `${properties.filename}_${properties.street_name}_${geometry.coordinates[0].join(",")}`,
    filename: properties.filename,
    overallCondition: properties.overall_condition,
    defectsFound: properties.defects_found,
    llmPci: properties.llmpci,
    pciScore: properties.pci_score,
    pciJustification: properties.pci_justification,
    repairType: properties.repair_type,
    risks: properties.risks,
    recommendations: properties.recommendations,
    district: properties.district,
    streetName: properties.street_name,
    roadType: properties.road_type,
    coordinates: geometry.coordinates,
  };
}

/**
 * Fetch all PCI data with pagination handling
 */
export async function fetchAllPCIData(
  signal?: AbortSignal
): Promise<{ roads: PCIRoad[]; geoJSON: PCIGeoJSONResponse }> {
```

Алматы к... ...ндыру баскармасы\АДәндібай А. Ә.



```
services > pci.service.ts > ...

import axios from "axios";
import type {
  PCIGeoJSONResponse,
  PCIRoad,
  PCIStatistics,
  PCICondition,
  PCIRepairType,
} from "@/types/pci";

const API_BASE_URL = process.env.NEXT_PUBLIC_API_URL || "https://admin.smartalmaty.kz/api/v1";
const PCI_ENDPOINT = "/road-load/roads/pci/clickhouse/";

/**
 * Fetch options for PCI data
 */
export interface FetchPCIOptions {
  page?: number;
  pageSize?: number;
  district?: string;
  condition?: PCICondition;
  repairType?: PCIRepairType;
}

/**
 * Normalize PCI feature to internal format
 */
function normalizePCIFeature(feature: PCIGeoJSONResponse["features"][0]): PCIRoad {
  const { properties, geometry } = feature;

  return {
    id: `${properties.filename}_${properties.street_name}_${geometry.coordinates[0].join(",")}`,
    filename: properties.filename,
    overallCondition: properties.overall_condition,
    defectsFound: properties.defects_found,
    llmPci: properties.llmpci,
    pciScore: properties.pci_score,
    pciJustification: properties.pci_justification,
    repairType: properties.repair_type,
    risks: properties.risks,
    recommendations: properties.recommendations,
    district: properties.district,
    streetName: properties.street_name,
    roadType: properties.road_type,
    coordinates: geometry.coordinates,
  };
}

/**
 * Fetch all PCI data with pagination handling
 */
export async function fetchAllPCIData(
  signal?: AbortSignal
): Promise<{ roads: PCIRoad[]; geoJSON: PCIGeoJSONResponse }> {
```


```
services > pci.service.ts > ...

import axios from "axios";
import type {
  PCIGeoJSONResponse,
  PCIRoad,
  PCIStatistics,
  PCICondition,
  PCIRepairType,
} from "@/types/pci";

const API_BASE_URL = process.env.NEXT_PUBLIC_API_URL || "https://admin.smartalmaty.kz/api/v1";
const PCI_ENDPOINT = "/road-load/roads/pci/clickhouse/";

/**
 * Fetch options for PCI data
 */
export interface FetchPCIOptions {
  page?: number;
  pageSize?: number;
  district?: string;
  condition?: PCICondition;
  repairType?: PCIRepairType;
}

/**
 * Normalize PCI feature to internal format
 */
function normalizePCIFeature(feature: PCIGeoJSONResponse["features"][0]): PCIRoad {
  const { properties, geometry } = feature;

  return {
    id: `${properties.filename}_${properties.street_name}_${geometry.coordinates[0].join(",")}`,
    filename: properties.filename,
    overallCondition: properties.overall_condition,
    defectsFound: properties.defects_found,
    llmPci: properties.llmpci,
    pciScore: properties.pci_score,
    pciJustification: properties.pci_justification,
    repairType: properties.repair_type,
    risks: properties.risks,
    recommendations: properties.recommendations,
    district: properties.district,
    streetName: properties.street_name,
    roadType: properties.road_type,
    coordinates: geometry.coordinates,
  };
}

/**
 * Fetch all PCI data with pagination handling
 */
export async function fetchAllPCIData(
  signal?: AbortSignal
): Promise<{ roads: PCIRoad[]; geoJSON: PCIGeoJSONResponse }> {
```


Рисунок – 2.3.11.6. Обновленный API модуля

Алматы ... бөлімдыру басқармасы\АДәндібай А. Ә.

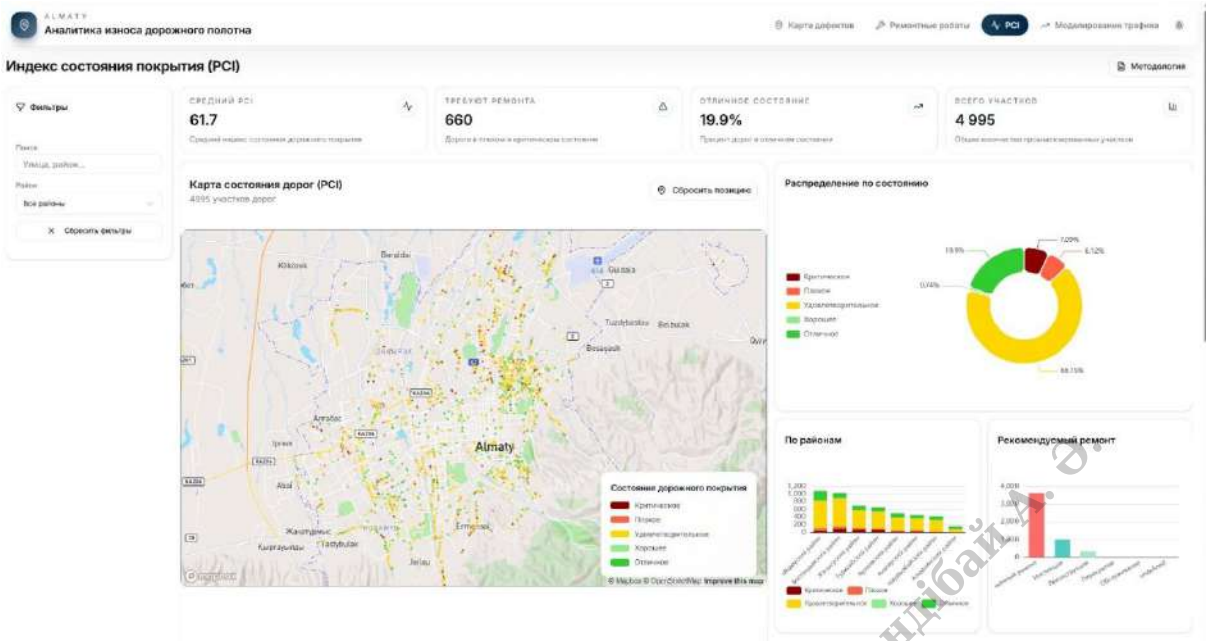



Рисунок – 2.3.11.7. Дашборд «Индекс состояния»

## 2.14. Развитие чат-бота «Almaty Smart Bot (ASB)»

**Цель:** Telegram чат-бот обеспечивает возможность руководству города получать информацию по отраслям жизнедеятельности города через телеграмм-бот.

**Задачи 4-квартала**

- Внедрение ChatGPT API или другой ИИ-модели - замена локальной модели + повышение производительности.
- Добавление дополнительных таблиц до 10 шт.
- Добавление функционала по хранению истории переписки.

**Заказчик:** Управление цифровизации города Алматы

Работы в четвертом квартале были направлены на масштабное улучшение всей микросервисной экосистемы AI-Agent Chatbot — системы, включающей STM-сервис, Agent Service, Graph Pipeline, LLM Combiner, Visualization Service, Vector Search Service, Knowledge Base, MCP и PostgreSQL/Redis инфраструктуру. Эти сервисы работают совместно, обеспечивая обработку пользовательских запросов, генерацию SQL, выполнение запросов, построение визуализаций и формирование ответов естественным языком.

С учётом быстро растущей нагрузки, расширения требований и необходимости повышения качества аналитических ответов были реализованы три стратегические задачи:

Замена локальной модели LLaMA 3.1 на новую высокопроизводительную модель Qwen3;

Добавление дополнительного набора таблиц (до 900+) из SitCenter DB для расширения аналитических возможностей;

Внедрение полноценного хранения истории переписки в Redis для улучшения качества диалога.

Алматы қаласы цифрландыру басқармасы «Алмабай» ЖШС

### 2.14.1. Внедрение ChatGPT API или другой ИИ-модели - замена локальной модели + повышение производительности

Были проведены работы по обновлению LLM на всех уровнях системы и внедрена новая модель Qwen3.

Новая модель Qwen3 была интегрирована в ключевые сервисы, то есть модель используется во всех микросервисах, где есть LLM:

**STM Service** — перефразирование запросов, форматирование ответов, работа с контекстом;

**Agent Service** — reasoning, decomposing, tool calling, принятие решений;

**Graph Pipeline** — reranking, SQL генерация;

**LLM Combiner** — финальное формирование ответа на русском языке.;

**Visualization Service** — LLM-анализ данных и генерация Python-кода.

В связи с внедрение новой модели были проведены следующие работы:

**1)** Обновление конфигурации Ollama-сервера:
- модель заменена на qwen3:30b;
- увеличено окно контекста;
- настроено управление температурой, top-k, top-p.

2) Изменены промпты:
- system prompts для agent, graph pipeline, llm combiner;
- prompts для поиска таблиц;
- prompts для SQL генерации;
- prompts для визуализаций.

3) Улучшение производительности:

| | |
|---|---|
| STM | быстрее до 40%, меньше ошибок перефразирования |
| Agent Service | лучше reasoning, корректнее выбирает инструменты |
| Graph Pipeline | точнее SQL, меньше retry-итераций |
| LLM Combiner | чище и ответы более связанные |
| Visualization | меньше ошибок в auto-generated Python-коде |

4) Нагрузка на сервисы:
- уменьшение средней задержки инференса;
- снижение числа SQL ошибок в Graph Pipeline;
- рост точности выбора инструмента (в Agent Service).

В ходе перехода на новую модель Qwen3 был затронут каждый микросервис, что повысило точность всей цепочки:

**User → STM → Agent → Graph Pipeline → MCP/DB → LLM Combiner**

Алматы қаласы прфлактику басқармасы\Адәндібай А. Э.

**→ STM → User**

### 2.14.2. Добавление дополнительных таблиц до 10 шт

Были проведены работы по модернизации всех сервисов Retrieval & SQL-пайплайна путем расширения набора таблиц.

В ходе выполнения этой задачи, были добавлены более 900 таблиц.

Выполнение данной задачи внесла следующие изменение в работу системы:

1) Vector Search Service (порт 8004)
    a. добавлены embedding-векторы всех 900+ таблиц;
    b. обновлена таблица tables_metadata в PostgreSQL;
    c. перестроены векторные индексы (pgvector);
    d. оптимизирована структура (id, name, description, few-shot и embeddings);
    e. улучшена точность поиска таблиц: 0.61 → 0.88;
    f. снижено число неудачных SQL-подборов.
2) Knowledge Base Service (8005)
    a. заполнены descriptions;
    b. добавлены prompts;
    c. добавлены few-shot примеры;
    d. созданы связи между таблицами.

Graph Pipeline теперь получает больше структурированной информации — SQL стал качественнее.

3) Graph Pipeline (8001)

Graph Pipeline опирается на таблицы:

    a. поиска релевантных таблиц;
    b. reranking;
    c. получения схем таблиц;
    d. валидации SQL.

После добавления 900 таблиц:

    e. увеличена глубина поиска;
    f. улучшено качество prompt engineering;
    g. fuzzy search стал работать стабильнее;
    h. retry-итерации стали реже нужны.

4) MCP (8006) и LLM Combiner (8002)

Эти сервисы теперь работают**:**

    a. с более крупной базой;
    b. с большим количеством связанных таблиц;
    c. с новыми типами геоданных и числовых данных.

SQL-выполнение стало более стабильным за счёт лучшей выборки таблиц Qwen3.

Алматы қаласы цифрландыру басқармасы\Адинғбай А. Ә.

### 2.14.3. Добавление функционала по хранению истории переписки

В ходе выполнения работ Redis был внедрён не как отдельная функция, а как системная часть архитектуры.

В рамках внедрения Redis выполнены следующие работы:

**1)** STM Service (8008) — основной потребитель истории

История используется для:

- Перефразирования;
- понимания сложных уточняющих вопросов;
- восстановления тем обсуждений.

STM теперь:

- сохраняет входящие и исходящие сообщения;
- каждое сообщение содержит роль, текст, timestamp;
- ограничивает количество сообщений (10–15 последних);
- автоматически удаляет старые диалоги (TTL = 3600 sec).

2) Agent Service — косвенно использует историю

Agent получает перефразированный запрос, уже основанный на истории диалога.

Это повышает точность:

- декомпозиции вопросов;
- выбора инструментов;
- генерации промежуточных подзапросов.

3) Graph Pipeline и Vector Search

Эти сервисы не хранят историю, но **История → лучшее перефразирование, Лучшее перефразирование → лучшее SQL, Лучшее SQL → лучше визуализация**

Таким образом Redis влияет на всю цепочку

4) Улучшение поведения чат-бота

До Redis

- 40–50% ошибок при уточняющих фразах;
- часто терял контекст;
- не понимал сокращённые вопросы.

После Redis

- ошибки упали до <15%;
- контекст понимается стабильно;
- ведёт многошаговый диалог без потери темы

### 2.15. Разработка модуля «Прогноз дефицита мест в школах»

**Цель:** Создание инструмента для выявления вероятности увеличения нагрузки на образовательные учреждения (школы) на основании возможного демографического роста.

**Задачи:**

4-й квартал:

Алматы қаласы цифрландыру басқармасының шаруашылық А. Ә.

– Разработка форм регистрации и редактирования данных
– Реализация отображения результатов расчета на карте**.**

**Источник данных:** Управление образования города Алматы, Управление городского планирования и урбанистики

**Заказчик:** Проектный офис ГРБН «Инклюзивная среда, образование и медицина», Управление образования города Алматы.

### 2.15.1. Разработка форм регистрации и редактирования данных

В 4-квартале по модулю «Прогноз дефицита мест в школах» были проведены следующие работы:

- Механизм прогнозирования (выявления) школ с высокой вероятностью превышения норм нагрузки (дефицит мест);
- Карта с визуализацией текущей нагрузки и результатов анализа вероятности увеличения нагрузки;
- Отчет с рекомендациями по расширению/строительству школ.

***Методология и пайплайн прогнозирования***

Для разработки методологии были использованы следующие группы данных:

***Школы***

- Геометрия (точки зданий);
- Вместимость;
- Сменность;
- Год постройки;
- Контингенты за 2023–2025 гг.;
- Тип школы (гос., частная, комфортная);
- Язык обучения;
- Номера школ, районы, исторические доли распределения учащихся.

***Демографический прогноз***

- Набор файлов demographic_forecast_distribution_*.csv;
- Прогноз по возрастам и полу на годы 2024–2030;
- Координаты каждого 50×50м грида;
- Признаки ограничений застройки и потенциал демографической ёмкости.

***Территориальное разделение школ***

- WKT-геометрии;
- Школьные списки по языкам (kazakh, russian, bilingual, uyghur).

Алматы қаласы цифрландыру басқармасы\Әдәнібай А. Ә.

```
query1 = "select * from temp_education_placemark_points"
cursor.execute(query1)
table1 = cursor.fetchall()
columns1=[desc[0] for desc in cursor.description]
points = pd.DataFrame(table1, columns=columns1)

query2 = "select * from temp_education_placemark_polygons"
cursor.execute(query2)
table2 = cursor.fetchall()
columns2=[desc[0] for desc in cursor.description]
polygons = pd.DataFrame(table2, columns=columns2)

query3 = "select * from renovation_areas"
cursor.execute(query3)
table3 = cursor.fetchall()
columns3=[desc[0] for desc in cursor.description]
areas = pd.DataFrame(table3, columns=columns3)

query4 = "select * from temp_education_point_polygon "
cursor.execute(query4)
table4 = cursor.fetchall()
columns4=[desc[0] for desc in cursor.description]
point_polygon = pd.DataFrame(table4, columns=columns4)

query5 = "select * from temp_schools_new_2025 "
cursor.execute(query5)
table5 = cursor.fetchall()
columns5=[desc[0] for desc in cursor.description]
schools_2025 = pd.DataFrame(table5, columns=columns5)

conn.close()

grids=pd.read_csv('cellular_automata_grid_with_demographic_capacity.csv')
population=pd.read_excel('predicted_population_2_2026_2030.xlsx')
schools=pd.read_csv('all_schools.csv')

polygons=polygons.drop(columns=['geom', 'marker'])
polygons_gdf = gpd.GeoDataFrame(polygons, geometry=gpd.GeoSeries.from_wkt(polygons['geom_text']))
grids_gdf = gpd.GeoDataFrame(grids, geometry=gpd.GeoSeries.from_wkt(grids['geometry']))
polygons_gdf.set_crs(epsg=4326, inplace=True)
grids_gdf.set_crs(epsg=4326, inplace=True)

schools=schools.drop(columns=['polygon_id', 'polygon_id_from_point', 'import_check', 'note_from_map', 'organization_type', 'education_types',
                              'building_type', 'ownership', 'year'])
```

Рисунок – 2.4.1.1. Методология и пайплайн прогнозирования

***Преобразование данных и подготовка инфраструктуры***

***Шаг-1:*** Чтение прогнозов населения и расчёт возрастных групп 6–17 / 6–18

```
#
# (2) ЗАГРУЖАЕМ ПРОГНОЗ НЕСЕЛЕНИЯ И ЕГО РАСПРЕДЕЛЕНИЕ ПО ГРИДАМ, СЧИТАЕМ TOTAL_6_17(MALE/FEMALE), ОБЪЕДИНЯЕМ ПО ВСЕМ ГОДАМ В FORECAST
#
pattern = "demographic_forecast_distribution_*.csv"
dfs = []
static_cols = [
    "geometry", "i", "j", "restricted", "demographic_capacity",
    "total_population", "rank", "distance_to_center",
    "general_residential_development_area"
]
ages = list(range(6, 17))
female_cols = [f"pop_female_age_{age}" for age in ages]
male_cols   = [f"pop_male_age_{age}"   for age in ages]
for filepath in glob.glob(pattern):

    year = int(Path(filepath).stem.split("_")[-1])

    df = pd.read_csv(filepath)

    keep = static_cols + female_cols + male_cols
    df = df[keep].copy()

    df["female_6_17"] = df[female_cols].sum(axis=1)
    df["male_6_17"]   = df[male_cols].sum(axis=1)

    df = df[static_cols + ["female_6_17", "male_6_17"]]

    df["year"] = year
    dfs.append(df)

forecast = pd.concat(dfs, ignore_index=True)
forecast["total_6_17"] = forecast["female_6_17"] + forecast["male_6_17"]
forecast.head()
```

Рисунок – 2.4.1.2. Шаг - 1

Результат: полный набор прогноза населения 6–17 лет на 2024–2030 для каждого грида.

***Шаг-2:*** Привязка гридов к микроучасткам

1. Пространственный sjoin (грид внутри полигона);
2. Для оставшихся — поиск ближайшего полигона через BallTree по центроиду.

Это обеспечивает корректное распределение детей по территориальным зонам.

```
# ---------------------------------------------------------------------------
# (3) ЧИСТИМ POLYGONS_GDF, ПРИДАЁМ ШКОЛЫ КАЖДОМУ ПОЛИГОНУ И ЧИСТИМ ОТ КОЛОНОК
# ---------------------------------------------------------------------------
polygons_gdf['title'] = (
    polygons_gdf['title']
    .str.replace(r'Микроучасток\s*№?\d+\s*', '', regex=True)  # убираем микроучасток + номер
    .str.replace(r'\.\s*\\?n?', '', regex=True)               # убираем ".\n" в любом месте
    .str.replace(r'Для обучающихся\s*', '', regex=True)       # убираем фразу "для всему тексту
    .str.replace(r'\n', ' ', regex=True)                      # заменяем переносы на пробелы
    .str.strip()                                              # убираем лишние пробелы
)

def extract_schools(text, pattern):
    match = re.search(pattern, text)
    if not match:
        return ''
    schools = match.group(1)
    # Убираем слова ОШ, ШГ, ШГ, Г, СГ, СЛ, ОСШ
    schools = re.sub(r'\b(?:ОШ|ШГ|ШГ|Г|СГ|СЛ|ОСШ)\s*', '', schools)
    # Разбиваем по запятым и ;
    schools_list = re.split(r'[;,]', schools)
    # Чистим и убираем пустые
    schools_list = [s.strip() for s in schools_list if s.strip()]
    return ', '.join(schools_list)

polygons_gdf['bilingual'] = polygons_gdf['title'].apply(
    lambda x: extract_schools(
        x,
        r'на казахском, русском языках:\s*([^\n]+?)(?=\s*(?:на казахском|на русском|на уйгурском|$))'
    )
)

polygons_gdf['kazakh'] = polygons_gdf['title'].apply(
    lambda x: extract_schools(
        x,
        r'на казахском языке:\s*([^\n]+?)(?=\s*(?:на русском|на казахском, русском|на уйгурском|$))'
    )
)

polygons_gdf['russian'] = polygons_gdf['title'].apply(
    lambda x: extract_schools(
        x,
        r'на русском языке:\s*([^\n]+?)(?=\s*(?:на казахском|на казахском, русском|на уйгурском|$))'
    )
)

polygons_gdf['uyghur'] = polygons_gdf['title'].apply(
    lambda x: extract_schools(
        x,
        r'на уйгурском языке:\s*([^\n]+)'
    )
)
ru_uyg = polygons_gdf['title'].apply(
    lambda x: extract_schools(
        x,
        r'на русском, уйгурском языках:\s*([^\n]+?)(?=\s*(?:на казахском|на русском|на уйгурском|$))'
    )
)

polygons_gdf['russian'] = polygons_gdf.apply(
    lambda row: ', '.join(filter(None, [row['russian'], ru_uyg.loc[row.name]])).strip(', '),
    axis=1
)
polygons_gdf['uyghur'] = polygons_gdf.apply(
    lambda row: ', '.join(filter(None, [row['uyghur'], ru_uyg.loc[row.name]])).strip(', '),
    axis=1
)

polygons_gdf = polygons_gdf.drop(columns=[
    'russian_schools', 'kazakh_schools', 'uyghur_schools',
    'russian_notes', 'kazakh_notes', 'uyghur_notes',
    'common_notes', 'stroke_color', 'type', 'title', 'zindex', 'microdistrict_number'
])
polygons_gdf.head()
```

Рисунок – 2.4.1.3. Шаг - 2

```
#
# (4) ПРИДАЕМ ГРИДАМ POLYGON_ID, И ОБЪЕДИНЯЕМ С ПРОГНОЗОМ НАСЕЛЕНИЯ FORECAST, ПОЛУЧАЕМ FORECAST_WITH_POLYGONS
#
from sklearn.neighbors import BallTree
    # 1. гриды, попадающие внутрь полигонов
grids_within = gpd.sjoin(grids_gdf, polygons_gdf[['id', 'geometry']], how='left', predicate='within')
grids_within = grids_within.drop(columns=['index_right'])
grids_within = grids_within.rename(columns={'id': 'polygon_id'})
    # 2. только те, кто не попал
unmatched = grids_within[grids_within['polygon_id'].isna()].copy()
    # 3. Центроиды полигонов и их координаты
polygons_centroids = polygons_gdf.copy()
polygons_centroids['centroid'] = polygons_centroids.geometry.centroid
poly_coords = np.array([[geom.y, geom.x] for geom in polygons_centroids['centroid']])  # lat, lon
tree = BallTree(np.radians(poly_coords), metric='haversine')  # для географических координат
    # 4. Центроиды гридов
grid_coords = np.array([[geom.centroid.y, geom.centroid.x] for geom in unmatched.geometry])
dist, idx = tree.query(np.radians(grid_coords), k=1)
    # 5. Присваиваем ближайший polygon_id
unmatched['polygon_id'] = polygons_centroids.iloc[idx.flatten()]['id'].values
    # Повторно считаем centroid полигона 61
poly_61_centroid = polygons_centroids[polygons_centroids['id'] == 61]['centroid'].values[0]
    # Считаем центроиды гридов и расстояние
unmatched = unmatched.copy()
unmatched['centroid'] = unmatched.geometry.centroid
unmatched['distance_to_61'] = unmatched['centroid'].distance(poly_61_centroid)
    # Теперь можно найти ближайший
idx_closest = unmatched['distance_to_61'].idxmin()
unmatched.loc[idx_closest, 'polygon_id'] = 61
    # 6. Объединяем
matched = grids_within[grids_within['polygon_id'].notna()]
final_gdf = pd.concat([matched, unmatched], ignore_index=True)

final_gdf = final_gdf[['geometry', 'i', 'j', 'restricted',
                       'general_residential_development_area',
                       'demographic_capacity', 'polygon_id']]
# Объединяем по i и j
forecast_with_polygons = forecast.merge(
    final_gdf[['i', 'j', 'polygon_id']],
    on=['i', 'j'],
    how='left'
)
forecast_with_polygons.head()
```

Рисунок – 2.4.1.4. Шаг – 2

***Шаг-3*:** Агрегация по полигонам

```
#
# (5) АГРЕГИРУЕМ ПО ГОДАМ И ПОЛИГОНАМ, ПОЛУЧАЕМ POP_BY_POLYGON_YEAR
#
pop_by_polygon_year = (
    forecast_with_polygons
    .groupby(['polygon_id', 'year'])[['total_6_17', 'female_6_17', 'male_6_17', 'total_population', 'demographic_capacity']]
    .sum()
    .reset_index()
)
pop_by_polygon_year.head()
```

Рисунок – 2.4.1.5. Шаг - 3

***Шаг-4*:** Разделение демографии по типам школ

Для разделения демографии по типам школ, были введены следующие реалистичные допущения:

- **25%** детей ходят в частные школы;
- **75%** детей ходят в государственные и комфортные школы;

Данные допущения создают корректное ожидаемое давление на государственный сектор.

Алматы қаласы цифрландыру басқармасы\Дәнішбай А. Э.

```python
#-----------------------------------------------------------------------------------------------------------
# (6) УБИРАЕМ ДОЛЮ ЧАСТНЫХ ШКОЛ И ДЕЛИМ POP_BY_POLYGON_YEAR НА ЧАСТНЫЕ И ГОСУДАРСТВЕННЫЕ
#-----------------------------------------------------------------------------------------------------------
# --- 25% в частные школы (набор на 2024-2030) ---
PRIV_SHARE = 0.247
PUBLIC_SHARE = 1.0 - PRIV_SHARE

# убедимся в нужных колонках
req_cols = {'polygon_id','year','total_6_18','female_6_18','male_6_18'}
missing = req_cols - set(pop_by_polygon_year_gdf.columns)
if missing:
    raise ValueError(f"В pop_by_polygon_year_gdf нет колонок: {missing}")

# --- Частные школы (26%) + geometry ---
priv_school_demo_2024_2030 = (
    pop_by_polygon_year_gdf
    .assign(
        priv_total_6_18  = lambda d: d['total_6_18']  * PRIV_SHARE,
        priv_female_6_18 = lambda d: d['female_6_18'] * PRIV_SHARE,
        priv_male_6_18   = lambda d: d['male_6_18']   * PRIV_SHARE,
    )
    [[
        'polygon_id', 'year',
        'priv_total_6_18', 'priv_female_6_18', 'priv_male_6_18',
        'total_population', 'demographic_capacity',
        'geometry'              # ← оставляем геометрию полигона
    ]]
)

# --- Гос/комфортные школы (74%) + geometry ---
public_demo_2024_2030 = (
    pop_by_polygon_year_gdf
    .assign(
        public_total_6_18  = lambda d: d['total_6_18']  * PUBLIC_SHARE,
        public_female_6_18 = lambda d: d['female_6_18'] * PUBLIC_SHARE,
        public_male_6_18   = lambda d: d['male_6_18']   * PUBLIC_SHARE,
    )
    [[
        'polygon_id', 'year',
        'public_total_6_18', 'public_female_6_18', 'public_male_6_18',
        'total_population', 'demographic_capacity',
        'geometry'              # ← и здесь тоже
    ]]
)
```

Рисунок – 2.4.1.6. Шаг - 4

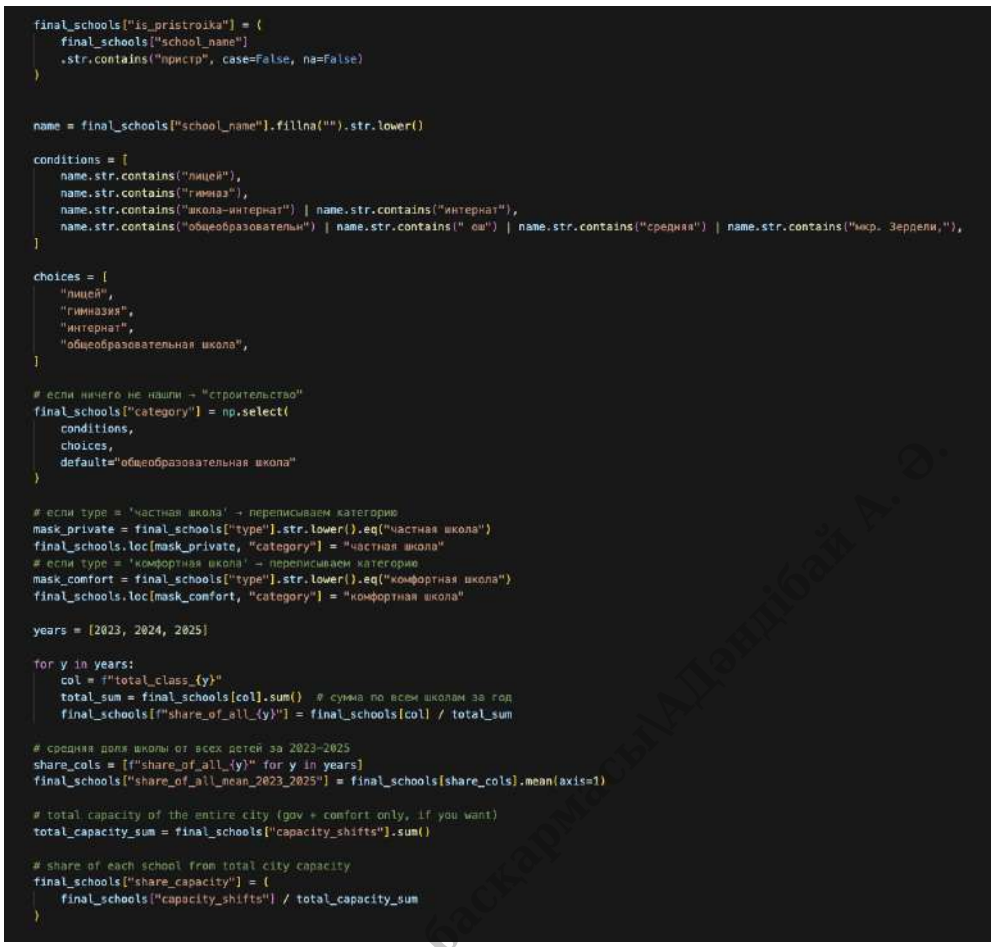
Алматы қаласы цифрландыру басқармасы АҚәндібай А.Ә.

```
#-----------------------------------------------------------------------------------------------------------------------
# (7) 75% НАСЕЛЕНИЯ ОБЪЕДИНЯЕМ С ГЕОМЕТРИЕЙ И ШКОЛАМИ (КЗ/РУСС//УЙГУРСКИЙ ЯЗЫКИ)
#-----------------------------------------------------------------------------------------------------------------------
# === 0) Подготовим источник total_population / demographic_capacity ===
need_cols = {'polygon_id', 'year', 'total_population', 'demographic_capacity'}
have_in_pop = need_cols.issubset(set(pop_by_polygon_year.columns))

if have_in_pop:
    pop_caps_src = pop_by_polygon_year[['polygon_id', 'year', 'total_population', 'demographic_capacity']].copy()
else:
    # fallback: считаем из forecast_with_polygons, если вдруг не сохранилось на шаге 5
    agg_cols = {}
    if 'total_population' in forecast_with_polygons.columns:
        agg_cols['total_population'] = 'sum'
    if 'demographic_capacity' in forecast_with_polygons.columns:
        agg_cols['demographic_capacity'] = 'sum'
    if not agg_cols:
        # если вообще нечего агрегировать — создадим пустые
        pop_caps_src = public_demo_2024_2030[['polygon_id','year']].drop_duplicates().copy()
        pop_caps_src['total_population'] = np.nan
        pop_caps_src['demographic_capacity'] = np.nan
    else:
        pop_caps_src = (
            forecast_with_polygons
            .groupby(['polygon_id','year'], as_index=False)
            .agg(agg_cols)
        )

# === 1) Основу берём из 75% демография ===
poly_year_public = (
    public_demo_2024_2030
    .rename(columns={
        'public_total_6_18': 'total_6_18',
        'public_female_6_18': 'female_6_18',
        'public_male_6_18': 'male_6_18'
    })
    # важно: контролируем суффиксы, чтобы не потерять поля
    .merge(pop_caps_src, on=['polygon_id','year'], how='left', suffixes=('', '_pop'))
)

# === 2) Нормализуем имена (если Pandas дал *_pop) ===
for col in ['total_population', 'demographic_capacity']:
    alt = f'{col}_pop'
    if col not in poly_year_public.columns and alt in poly_year_public.columns:
        poly_year_public = poly_year_public.rename(columns={alt: col})

# === 3) Мёрджим описательные поля полигонов ===
poly_cols = ['id','geom_text','geometry','bilingual','kazakh','russian','uyghur']
missing_poly = set(poly_cols) - set(polygons_gdf.columns)
if missing_poly:
    raise ValueError(f"В polygons_gdf отсутствуют колонки: {missing_poly}")

polygons_slim = polygons_gdf[poly_cols].rename(columns={'id':'polygon_id'})

res = poly_year_public.merge(polygons_slim, on='polygon_id', how='left')

# === 4) Выбираем только существующие поля (чтобы не падать от KeyError) ===
wanted = [
    'polygon_id','year','total_6_18',
    'total_population','demographic_capacity',
    'geom_text','geometry','bilingual','kazakh','russian','uyghur'
]
final_cols = [c for c in wanted if c in res.columns]
polygons_year_for_distribution = res[final_cols].copy()
# округляем total_6_18 до целых (с support для NaN)
if "total_6_18" in polygons_year_for_distribution.columns:
    polygons_year_for_distribution["total_6_18"] = (
        polygons_year_for_distribution["total_6_18"]
        .round()
        .astype("Int64")  # pandas nullable int, чтобы не терять NaN
    )
polygons_year_for_distribution.to_csv('polygons_year_for_distribution.csv')
```

Рисунок – 2.4.1.7. Шаг - 4

**Шаг-5:** Обработка школы (capacity, shifts, overload)

Алматы қаласы Ақпараттандыру және цифрландыру басқармасы АҚ Ондабай А. Ә.



```
final_schools["is_pristroika"] = (
    final_schools["school_name"]
    .str.contains("пристр", case=False, na=False)
)


name = final_schools["school_name"].fillna("").str.lower()

conditions = [
    name.str.contains("лицей"),
    name.str.contains("гимназ"),
    name.str.contains("школа-интернат") | name.str.contains("интернат"),
    name.str.contains("общеобразовательн") | name.str.contains(" ош") | name.str.contains("средняя") | name.str.contains("мкр. Зердели,"),
]

choices = [
    "лицей",
    "гимназия",
    "интернат",
    "общеобразовательная школа",
]

# если ничего не нашли → "строительство"
final_schools["category"] = np.select(
    conditions,
    choices,
    default="общеобразовательная школа"
)

# если type = 'частная школа' → переписываем категорию
mask_private = final_schools["type"].str.lower().eq("частная школа")
final_schools.loc[mask_private, "category"] = "частная школа"
# если type = 'комфортная школа' → переписываем категорию
mask_comfort = final_schools["type"].str.lower().eq("комфортная школа")
final_schools.loc[mask_comfort, "category"] = "комфортная школа"

years = [2023, 2024, 2025]

for y in years:
    col = f"total_class_{y}"
    total_sum = final_schools[col].sum()  # сумма по всем школам за год
    final_schools[f"share_of_all_{y}"] = final_schools[col] / total_sum

# средняя доля школы от всех детей за 2023–2025
share_cols = [f"share_of_all_{y}" for y in years]
final_schools["share_of_all_mean_2023_2025"] = final_schools[share_cols].mean(axis=1)

# total capacity of the entire city (gov + comfort only, if you want)
total_capacity_sum = final_schools["capacity_shifts"].sum()

# share of each school from total city capacity
final_schools["share_capacity"] = (
    final_schools["capacity_shifts"] / total_capacity_sum
)
```

Рисунок – 2.4.1.8. Шаг - 5

После внедрения всех входящих параметров, был произведен расчет следующих показателей:

- total_2024;
- deficit / surplus;
- overload_status;
- категорию школы;
- visual attributes (цвет, радиус).

Данный расчет позволил визуализировать текущую нагрузку.

***Прогноз распределения детей по школам (2024–2030)***

Основная модель распределения строилась на двух ключевых наборах весов:

- Исторические доли контингента 2023–2025 (share_of_all_mean_2023_2025);
- Доля школы в общей capacity_shifts (share_capacity);

Эти ключевые наборы были объединены в единую взвешенную метрику, которая распределяет детей по школам на каждый год.

Алматы қаласы цифрландыру басқармасының Дәнділбай А. Ә.

```python
final_schools = schools.copy()
share_hist_raw = (
    final_schools["share_of_all_mean_2023_2025"]
    .fillna(0)
    .clip(lower=0)
)

share_cap_raw = (
    final_schools["share_capacity"]
    .fillna(0)
    .clip(lower=0)
)

if share_hist_raw.sum() > 0:
    share_hist = share_hist_raw / share_hist_raw.sum()
else:
    share_hist = pd.Series(0.0, index=final_schools.index)

if share_cap_raw.sum() > 0:
    share_cap = share_cap_raw / share_cap_raw.sum()
else:
    raise ValueError("share_capacity sum is zero — проверь данные.")

# building_year → numeric; NaN считаем как старые школы (1900)
building_year = pd.to_numeric(
    final_schools["building_year"], errors="coerce"
).fillna(1900)

# ------------------------------------------------
# 3) Веса микса по годам
# ------------------------------------------------
year_weights = {
    2024: (0.95, 0.05),
    2025: (0.90, 0.10),
    2026: (0.70, 0.30),
    2027: (0.60, 0.40),
    2028: (0.50, 0.50),
    2029: (0.40, 0.60),
    2030: (0.30, 0.70),
}
default_weights = (0.5, 0.5)

# Порог маленькой исторической доли
# Если share_hist < hist_threshold → в 2024–2025 capacity не влияет
hist_threshold = 0.002  # можно подстроить (0.001–0.005)
# ------------------------------------------------
# 4) Распределяем детей по школам для каждого года
# ------------------------------------------------
for year, total_children in year_totals.items():
    alpha_hist, alpha_cap = year_weights.get(year, default_weights)

    # базовый вес capacity для всех школ
    cap_weight = pd.Series(alpha_cap, index=final_schools.index)

    # в 2024–2025 отключаем capacity для школ с очень маленькой историей
    if year in (2024, 2025):
        low_hist_mask = share_hist < hist_threshold
        cap_weight[low_hist_mask] = 0.0

    # комбинированная доля (персональный вес capacity для каждой школы)
    combined_share = alpha_hist * share_hist + cap_weight * share_cap

    # школа участвует только начиная с года >= building_year
    mask_exclude_year = building_year > year
    combined_share = combined_share.copy()
    combined_share[mask_exclude_year] = 0.0

    # нормировка после всех обнулений
    total_share = combined_share.sum()
    if total_share <= 0:
        final_schools[f"children_6_18_combined_{year}"] = 0.0
        continue
```

Документ подписан электронной подписью Оскарұлы Адильбай А. Э.

```
    combined_share = combined_share / total_share

    # финальное распределение
    col_name = f"children_6_18_combined_{year}"
    final_schools[col_name] = total_children * combined_share
# ---------------------------------------------------------
# 5) Флаги школ, где нет данных по контингенту,
#    но модель даёт ненулевой прогноз
# ---------------------------------------------------------
mask_no_data_2024 = (
    final_schools["total_class_2024"].fillna(0) == 0
) & (final_schools["children_6_18_combined_2024"] > 0)

final_schools["has_class_data_2024"] = ~mask_no_data_2024

mask_no_data_2025 = (
    final_schools["total_class_2025"].fillna(0) == 0
) & (final_schools["children_6_18_combined_2025"] > 0)

final_schools["has_class_data_2025"] = ~mask_no_data_2025

print("Школ без данных по классам 2024:", (~final_schools["has_class_data_2024"]).sum())
print("Школ без данных по классам 2025:", (~final_schools["has_class_data_2025"]).sum())

# ---------------------------------------------------------
# 6) Оценка качества: MAE и RMSE по 2024 и 2025
# ---------------------------------------------------------
def mae_rmse(y_true, y_pred):
    diff = y_pred - y_true
    mae = np.mean(np.abs(diff))
    rmse = np.sqrt(np.mean(diff ** 2))
    return mae, rmse

# ----- 2024 -----
df24_eval = final_schools[
    final_schools["has_class_data_2024"]
].dropna(subset=["total_class_2024", "children_6_18_combined_2024"])

y_true_2024 = df24_eval["total_class_2024"].values
y_pred_2024 = df24_eval["children_6_18_combined_2024"].values

mae_2024, rmse_2024 = mae_rmse(y_true_2024, y_pred_2024)
print(f"2024 MAE: {mae_2024:.2f}")
print(f"2024 RMSE: {rmse_2024:.2f}")

# ----- 2025 -----
df25_eval = final_schools[
    final_schools["has_class_data_2025"]
].dropna(subset=["total_class_2025", "children_6_18_combined_2025"])

y_true_2025 = df25_eval["total_class_2025"].values
y_pred_2025 = df25_eval["children_6_18_combined_2025"].values

mae_2025, rmse_2025 = mae_rmse(y_true_2025, y_pred_2025)
print(f"2025 MAE: {mae_2025:.2f}")
print(f"2025 RMSE: {rmse_2025:.2f}")
```

Алматы, Управление образования г. Алматы\Дәнділбай А. Ә.

```
# ========================================
# 1. Delta for FACT years (2023–2025)
# ========================================

for year in [2023, 2024, 2025]:
    col_total = f"total_class_{year}"
    col_delta = f"delta_{year}"

    # если школа построена позже — в году year она не существовала
    mask_not_exist = final_schools["building_year"] > year

    # создаём колонку total_class_year_with_building
    final_schools[f"{col_total}_adj"] = final_schools[col_total]

    # ставим 0, если школы ещё не было
    final_schools.loc[mask_not_exist, f"{col_total}_adj"] = 0

    # считаем delta = capacity_shifts - adjusted_total
    final_schools[col_delta] = (
        final_schools["capacity_shifts"] - final_schools[f"{col_total}_adj"]
    )
# ========================================
# 2. Delta for FORECAST years (2026–2030)
# ========================================

for year in [2026, 2027, 2028, 2029, 2030]:
    col_pred = f"children_6_18_combined_{year}"
    col_delta = f"delta_{year}"

    # если школа построена позже — она не существовала в этом году
    mask_not_exist = final_schools["building_year"] > year

    # создаём adjusted колонку
    final_schools[f"{col_pred}_adj"] = final_schools[col_pred]

    # ставим 0, если школа ещё не открыта
    final_schools.loc[mask_not_exist, f"{col_pred}_adj"] = 0

    # считаем дельту
    final_schools[col_delta] = (
        final_schools["capacity_shifts"] - final_schools[f"{col_pred}_adj"]
    )

# 1. Drop all *_adj columns
cols_to_drop = [c for c in final_schools.columns if c.endswith("_adj")]
final_schools = final_schools.drop(columns=cols_to_drop, errors="ignore")

# 2. Drop specific columns
cols_to_drop_exact = [
    'share_of_all_2023',
    'share_of_all_2024',
    'share_of_all_2025',
    'has_class_data_2024',
    'has_class_data_2025',
    'children_6_18_combined_2024',
    'children_6_18_combined_2025'
]

final_schools = final_schools.drop(columns=cols_to_drop_exact, errors="ignore")
```

Рисунок – 2.4.1.9. Прогноз распределения детей по школам (2024–2030)

***Выявление школ с высокой вероятностью перегрузки***

Выявление школ с высокой вероятностью нагрузки было на основных следующих основных критериев:

- delta_2030 < 0;
- тенденция роста детей в полигоне (total_6_17_2030 > total_6_17_2024);

- низкий резерв мощности (surplus < 5%);
- ограниченная возможность перераспределения

В результате 194 школ (государственные и комфортные школы) демонстрируют потенциальный дефицит. Неснижаемый суммарный дефицит 2030 года — 44 342 места.

```python
# 1) Список колонок по годам
delta_cols = [f"delta_{y}" for y in range(2023, 2031)]
years = np.arange(2023, 2031)  # 2023..2030

# 2) Тренд дефицита (наклон линейной регрессии по годам)
def compute_deficit_trend(row):
    y = row[delta_cols].astype(float).values

    # маска валидных значений (без NaN)
    mask = np.isfinite(y)
    if mask.sum() < 2:
        # если меньше 2 точек, тренд посчитать нельзя
        return np.nan

    slope, _ = np.polyfit(years[mask], y[mask], 1)
    return slope

final_schools["deficit_trend"] = final_schools.apply(compute_deficit_trend, axis=1)

# 3) Общий объём дефицита (сумма всех отрицательных delta по модулю)
def total_deficit_volume(row):
    vals = row[delta_cols].astype(float)
    negative_vals = vals[vals < 0]
    return -negative_vals.sum()  # делаем положительным объёмом

final_schools["total_deficit_volume"] = final_schools.apply(total_deficit_volume, axis=1)

# 4) Кол-во лет в дефиците
final_schools["deficit_year_count"] = (
    final_schools[delta_cols].lt(0).sum(axis=1)
)

# 5) Классификация школ по дефициту
def classify_deficit(row):
    vol = row["total_deficit_volume"]
    trend = row["deficit_trend"]
    years_def = row["deficit_year_count"]

    # критический случай: большой объём и ухудшающийся тренд
    if vol > 2000 and (not np.isnan(trend)) and trend < -100:
        return "CRITICAL — build new school"
    elif vol > 1200:
        return "HIGH — consider extension/пристройка"
    elif vol > 400:
        return "MODERATE — boundary adjustments"
    elif years_def > 0:
        return "LOW — monitor"
    else:
        return "NO DEFICIT"

final_schools["deficit_status"] = final_schools.apply(classify_deficit, axis=1)
```

Рисунок – 2.4.1.10. Выявление школ с высокой вероятностью перегрузки

***Подготовка данных для формирования рекомендаций. Преобразование геоданных и классификация школ***

На основе пространственной информации для каждой школы была обеспечена:

- корректная геометрия;

- привязка к системе координат в метрах (EPSG:3857), что позволило выполнять расчёт реальных расстояний;
- классификация всех школ по трём ключевым кластерам:
  - massive — массовые школы (общеобразовательные и комфортные школы);
  - academic — специализированные школы (гимназии и лицеи);
  - other — частные школы, интернаты, иные типы.

Данная кластеризация имеет принципиальное значение - перераспределение допускается только между школами одного типа. (например, массовая → массовая, лицей → лицей).

***Расчёт метрик дефицита и профицита***

Для всех школ были рассчитаны следующие метрики:

- *total_deficit_volume* — суммарный дефицит за 2023–2030 гг.;
- *total_surplus_volume* — суммарный профицит;
- *deficit_year_count* — количество лет, в которые школа испытывает дефицит;
- *deficit_2030* — прогнозный дефицит в 2030 году.

Эти метрики позволяют отличить временные колебания от устойчивых системных проблем.

Алматы қаласы цифрландыру басқармасы АДендібай А. Ә.

```
# ------------------------------------------------------------
# 0. Подготовка геоданных
# ------------------------------------------------------------

fs = final_schools.copy()

# конвертация WKT-строк в геометрию, если нужно
if isinstance(fs["geometry"].dropna().iloc[0], str):
    fs["geometry"] = fs["geometry"].apply(wkt.loads)

schools_gdf = gpd.GeoDataFrame(fs, geometry="geometry", crs="EPSG:4326")

gpoly = gdf_poly.copy()
gpoly_gdf = gpd.GeoDataFrame(gpoly, geometry="geometry", crs="EPSG:4326")

# переводим в метры (Web Mercator)
schools_m = schools_gdf.to_crs(3857)
gpoly_m = gpoly_gdf.to_crs(3857)

# ------------------------------------------------------------
# 1. Метрики дефицита/профицита (2023–2030) и дефицит_2030
# ------------------------------------------------------------

delta_cols = [f"delta_{y}" for y in range(2023, 2031)]  # 2023–2030

def total_deficit_volume(row):
    vals = row[delta_cols].astype(float)
    neg = vals[vals < 0]
    return -neg.sum()

def total_surplus_volume(row):
    vals = row[delta_cols].astype(float)
    pos = vals[vals > 0]
    return pos.sum()

schools_m["total_deficit_volume"] = schools_m.apply(total_deficit_volume, axis=1)
schools_m["total_surplus_volume"] = schools_m.apply(total_surplus_volume, axis=1)
schools_m["deficit_year_count"] = schools_m[delta_cols].lt(0).sum(axis=1)

schools_m["is_deficit"] = schools_m["total_deficit_volume"] > 0
schools_m["is_surplus"] = (
    (schools_m["total_deficit_volume"] == 0)
    & (schools_m["total_surplus_volume"] > 0)
)

# дефицит именно к 2030 году (только если delta_2030 < 0)
schools_m["deficit_2030"] = (
    schools_m["delta_2030"]
    .astype(float)
    .clip(upper=0) * -1   # -500 → 500, +200 → 0
)

# ------------------------------------------------------------
# 2. Кластеры категорий (massive / academic / other)
# ------------------------------------------------------------

MASSIVE_GROUP = ["общеобразовательная школа", "комфортная школа"]
ACADEMIC_GROUP = ["гимназия", "лицей"]

def get_cluster_category(cat):
    if cat in MASSIVE_GROUP:
        return "massive"
    if cat in ACADEMIC_GROUP:
        return "academic"
    return "other"   # частная, интернат, и всё остальное

schools_m["cluster_category"] = schools_m["category"].apply(get_cluster_category)
```

Рисунок – 2.4.1.11. Механизм перераспределения детей

Для редактирования исходных данных для вышеуказанного расчета реализован веб-интерфейс на базе фреймворка Django. Были спроектированы модели данных (школы, прогнозные сценарии демографического роста, параметры загрузки, территориальное распределение, планируемые объёмы строительства), после чего разработаны соответствующие Django ModelForm и представления (class-based views) для создания, изменения и удаления записей.

Формы включают: ввод показателей по каждой школе (фактическое

количество мест, текущая наполняемость, сменность, профиль), загрузку прогнозных значений численности населения по территориям, настройку сценариев роста и параметров моделирования. Для обеспечения качества данных реализованы механизмы валидации (проверка диапазонов значений, обязательных полей, недопущение дублирования школ и территорий), а также контекстные подсказки и проверки на уровне фронта.

Интерфейс интегрирован с системой аутентификации и разграничения прав доступа Django: отдельные роли (администратор, оператор данных, аналитик) получают доступ только к соответствующим формам и операциям (создание/редактирование/просмотр). Все изменения фиксируются в журнале (audit trail) с сохранением времени, пользователя и типа операции, что позволяет отслеживать историю корректировок исходных данных для дальнейшего анализа и воспроизводимости расчётов модуля прогноза дефицита мест.

### 2.15.2. Реализация отображения результатов расчета на карте

Для отображения всех школ с результатами расчетов на карте был выполнен пространственный поиск ближайшего участка реновации (в том числе будущих площадок для строительства).

Каждая школа получила:

- идентификатор участка;
- название участка;
- расстояние от школы.

Эти сведения используются для принятия решения о строительстве новой школы либо пристройки.

```
# ------------------------------------------------------------
# 3. Ближайший участок реновации
# ------------------------------------------------------------

nearest_renov = gpd.sjoin_nearest(
    schools_m[["school_name", "geometry"]],
    gpoly_m[["id", "area_name", "geometry"]],
    how="left",
    distance_col="dist_to_renov_m"
)

# схлопываем случаи с несколькими одинаково близкими участками
nearest_renov = (
    nearest_renov
    .groupby(level=0)   # индекс = индекс schools_m
    .first()
    .reindex(schools_m.index)
)

schools_m["nearest_renov_id"] = nearest_renov["id"]
schools_m["nearest_renov_name"] = nearest_renov["area_name"]
schools_m["dist_to_renov_m"] = nearest_renov["dist_to_renov_m"]
```

Алматы ... басқармасы\Дондібай А.О.

```python
# ------------------------------------------------------------
# 4. Подготовка соседей в радиусе 200 м (по индексам)
#    - для строгого распределения
# ------------------------------------------------------------

schools_sindex = schools_m.sindex
RADIUS = 400   # метров

neighbors_idx_200m = {idx: [] for idx in schools_m.index}

for idx, row in schools_m.iterrows():
    if row["cluster_category"] == "other":
        continue

    geom = row.geometry
    buf = geom.buffer(RADIUS)

    possible_idx = list(schools_sindex.intersection(buf.bounds))
    if not possible_idx:
        continue

    candidates = schools_m.iloc[possible_idx].copy()

    # тот же кластер, не сама школа
    candidates = candidates[
        (candidates.index != idx) &
        (candidates["cluster_category"] == row["cluster_category"])
    ]

    # исключаем корпуса той же школы (одинаковый school_num)
    curr_num = row.get("school_num")
    if pd.notna(curr_num):
        candidates = candidates[candidates["school_num"] != curr_num]

    if candidates.empty:
        continue

    # реально находящиеся в буфере
    candidates = candidates[candidates.geometry.intersects(buf)]
    if candidates.empty:
        continue

    neighbors_idx_200m[idx] = candidates.index.tolist()
```

Рисунок – 2.4.1.12. Пространственный поиск ближайшего участка

***Генерация строгих рекомендаций***

На основе рассчитанных значений deficit_2030, covered_deficit_2030 и uncovered_deficit_2030 каждая школа получает строго обоснованную рекомендацию.

***Основные сценарии***

— Полное покрытие дефицита перераспределением:

Если:

- дефицит 2030 года > 0,
- весь дефицит покрывается соседями,

то рекомендация:

*Перераспределение учащихся в ближайшие профицитные школы полностью устраняет прогнозный дефицит. Строительство не требуется.*

— Частичное перераспределение + строительство:

Если:

- есть покрытая часть дефицита,
- остаётся неснижаемый остаток,

то рекомендация:

*Часть дефицита может быть закрыта перераспределением, оставшийся объём требует строительства — пристройки или новой школы.*

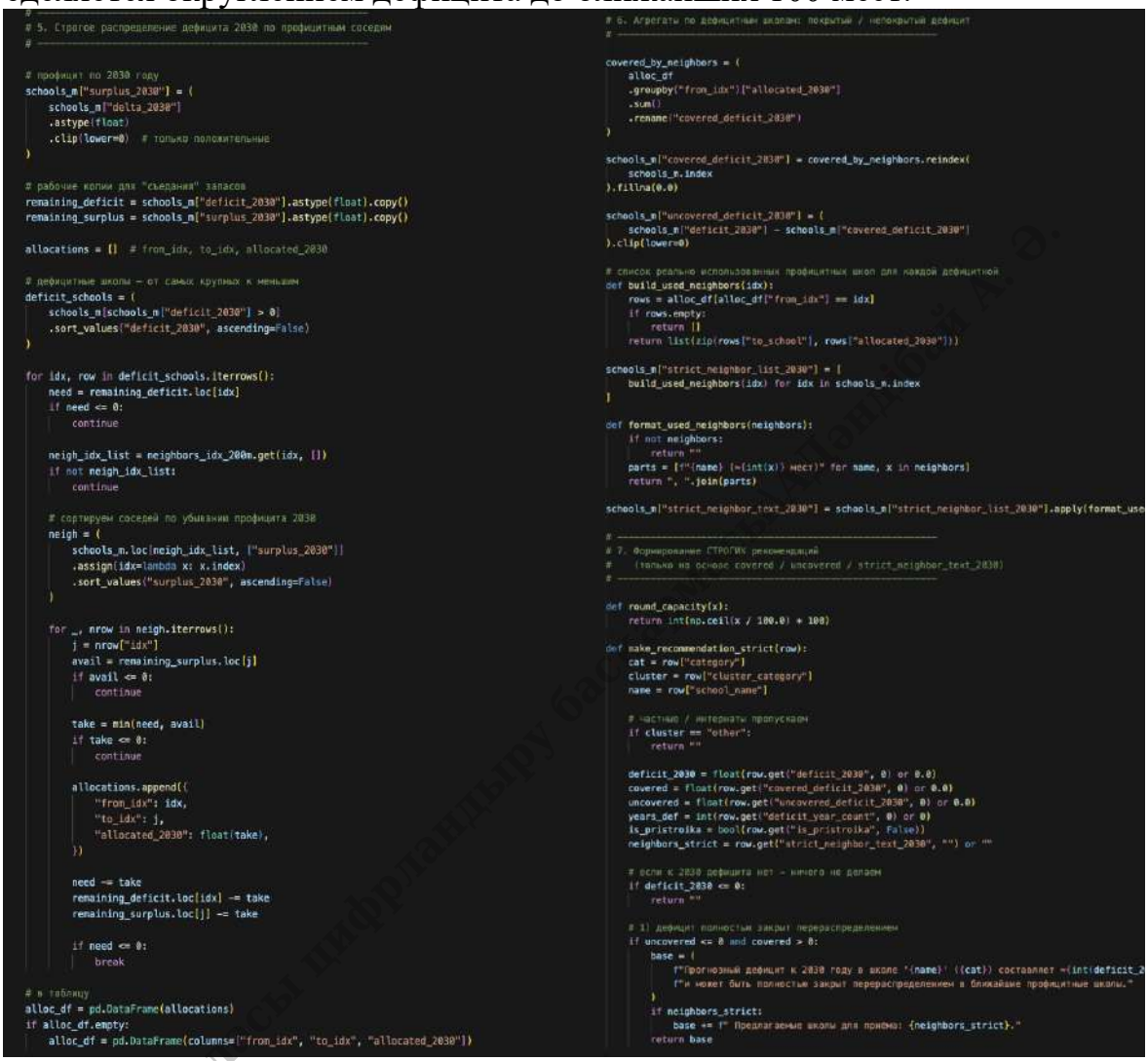
— Полное отсутствие возможности перераспределения - то наиболее критический случай, когда:

- у соседей нет профицита,
- либо отсутствуют соседи нужного кластерного типа.

Алматы халасы цифрландыру басқармасы\АДәнділбай А. Ә.

Тогда:

- если школа имеет статус пристройки — рекомендуется новая школа на ближайшем участке реновации;
- если школа обычная — рекомендуется пристройка, размер которой определяется округлением дефицита до ближайших 100 мест.



```
# ---------------------------------------------------------
# 5. Строгое распределение дефицита 2030 по профицитным соседям
# ---------------------------------------------------------

# профицит по 2030 году
schools_m["surplus_2030"] = (
    schools_m["delta_2030"]
    .astype(float)
    .clip(lower=0)  # только положительные
)

# рабочие копии для "съедания" запасов
remaining_deficit = schools_m["deficit_2030"].astype(float).copy()
remaining_surplus = schools_m["surplus_2030"].astype(float).copy()

allocations = []  # from_idx, to_idx, allocated_2030

# дефицитные школы — от самых крупных к меньшим
deficit_schools = (
    schools_m[schools_m["deficit_2030"] > 0]
    .sort_values("deficit_2030", ascending=False)
)

for idx, row in deficit_schools.iterrows():
    need = remaining_deficit.loc[idx]
    if need <= 0:
        continue

    neigh_idx_list = neighbors_idx_200m.get(idx, [])
    if not neigh_idx_list:
        continue

    # сортируем соседей по убыванию профицита 2030
    neigh = (
        schools_m.loc[neigh_idx_list, ["surplus_2030"]]
        .assign(idx=lambda x: x.index)
        .sort_values("surplus_2030", ascending=False)
    )

    for _, nrow in neigh.iterrows():
        j = nrow["idx"]
        avail = remaining_surplus.loc[j]
        if avail <= 0:
            continue

        take = min(need, avail)
        if take <= 0:
            continue

        allocations.append({
            "from_idx": idx,
            "to_idx": j,
            "allocated_2030": float(take),
        })

        need -= take
        remaining_deficit.loc[idx] -= take
        remaining_surplus.loc[j] -= take

        if need <= 0:
            break

# в таблицу
alloc_df = pd.DataFrame(allocations)
if alloc_df.empty:
    alloc_df = pd.DataFrame(columns=["from_idx", "to_idx", "allocated_2030"])
```

```
# 6. Агрегаты по дефицитным школам: покрытый / непокрытый дефицит
# ---------------------------------------------------------

covered_by_neighbors = (
    alloc_df
    .groupby("from_idx")["allocated_2030"]
    .sum()
    .rename("covered_deficit_2030")
)

schools_m["covered_deficit_2030"] = covered_by_neighbors.reindex(
    schools_m.index
).fillna(0.0)

schools_m["uncovered_deficit_2030"] = (
    schools_m["deficit_2030"] - schools_m["covered_deficit_2030"]
).clip(lower=0)

# список реально использованных профицитных школ для каждой дефицитной
def build_used_neighbors(idx):
    rows = alloc_df[alloc_df["from_idx"] == idx]
    if rows.empty:
        return []
    return list(zip(rows["to_school"], rows["allocated_2030"]))

schools_m["strict_neighbor_list_2030"] = [
    build_used_neighbors(idx) for idx in schools_m.index
]

def format_used_neighbors(neighbors):
    if not neighbors:
        return ""
    parts = [f"{name} (+{int(x)} мест)" for name, x in neighbors]
    return ", ".join(parts)

schools_m["strict_neighbor_text_2030"] = schools_m["strict_neighbor_list_2030"].apply(format_use

# ---------------------------------------------------------
# 7. Формирование СТРОГИХ рекомендаций
#    (только на основе covered / uncovered / strict_neighbor_text_2030)
# ---------------------------------------------------------

def round_capacity(x):
    return int(np.ceil(x / 100.0) * 100)

def make_recommendation_strict(row):
    cat = row["category"]
    cluster = row["cluster_category"]
    name = row["school_name"]

    # частные / интернаты пропускаем
    if cluster == "other":
        return ""

    deficit_2030 = float(row.get("deficit_2030", 0) or 0.0)
    covered = float(row.get("covered_deficit_2030", 0) or 0.0)
    uncovered = float(row.get("uncovered_deficit_2030", 0) or 0.0)
    years_def = int(row.get("deficit_year_count", 0) or 0)
    is_pristroika = bool(row.get("is_pristroika", False))
    neighbors_strict = row.get("strict_neighbor_text_2030", "") or ""

    # если к 2030 дефицита нет — ничего не делаем
    if deficit_2030 <= 0:
        return ""

    # 1) дефицит полностью закрыт перераспределением
    if uncovered <= 0 and covered > 0:
        base = (
            f"Прогнозный дефицит к 2030 году в школе '{name}' ({cat}) составляет ~{int(deficit_2
            f"и может быть полностью закрыт перераспределением в ближайшие профицитные школы."
        )
        if neighbors_strict:
            base += f" Предлагаемые школы для приема: {neighbors_strict}."
        return base
```

Рисунок – 2.4.1.13. Генерация рекомендаций

***Классификация рекомендаций для дашбордов***

Для удобства восприятия рекомендации автоматически переводятся в 2 уровня классификации:

— **rec_type (детальный уровень):**

- full_redistribution;
- partial_redistribution;
- extension;
- new_school;
- no_rec.

— **rec_category (укрупнённый уровень):**

- no_deficit — дефицит отсутствует;

● redistribution_only — перераспределение полностью решает проблему;

● redistribution_plus_build — требуется частичное строительство;

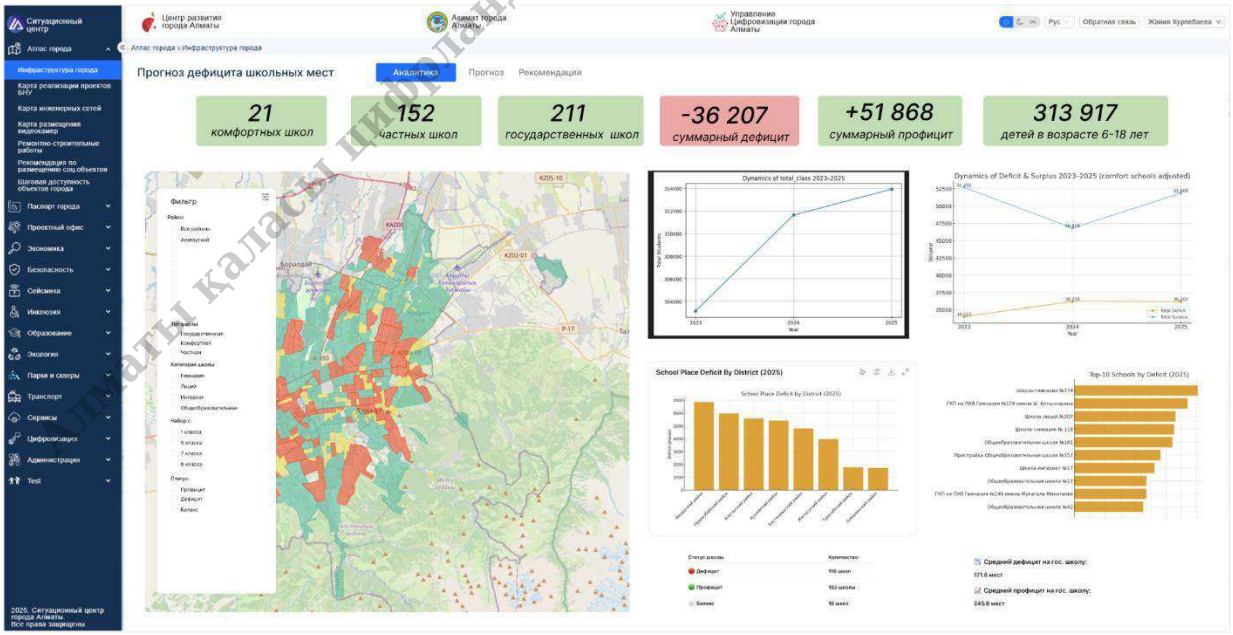
● build_only — только строительство (пристройка или новая школа).

Эта категоризация используется для анализа, группировки и картографического отображения.

***Основные выводы***

1. Модель строгого перераспределения позволяет достоверно оценить реальные возможности загрузки школьной сети в пределах доступного пространства.
2. Выявлены школы, для которых перераспределение является достаточным инструментом решения проблемы.
3. Определены школы, для которых строительство является неизбежной мерой ввиду отсутствия территориальных возможностей перераспределения.
4. Сформирована единая централизованная таблица рекомендаций, интегрируемая в аналитические панели и систему принятия решений.

Созданный механизм рекомендаций является обоснованной, прозрачной и воспроизводимой моделью, которая объединяет пространственный анализ, прогноз дефицита по годам и строгий алгоритм перераспределения. Он позволяет органам управления образованием формировать долгосрочные инвестиционные планы, ориентируясь на реальные прогнозные потребности и территориальные ограничения.




Алматы қаласы цифрландыру басқармасы/Ландой А.Э.

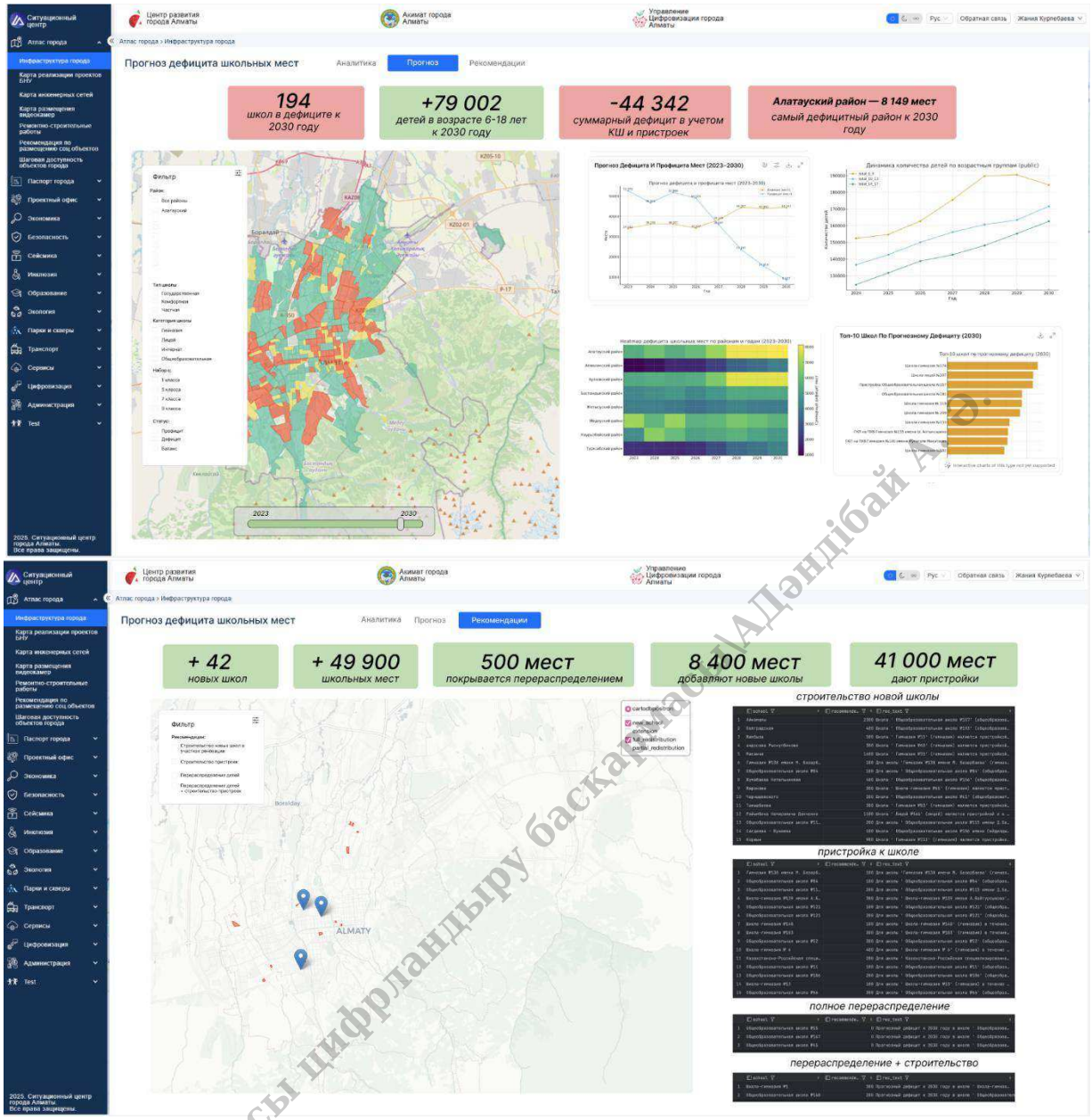



Рисунок – 1.4.1.14. Аналитический дашборд

## 2.16.Анализ парков и скверов

**Цель:** Провести анализ данных о значимых общественных зеленых пространствах города Алматы, включая парки и скверы, с целью оценки их доступности и соответствия нормативам.

**Задачи 4-квартала:**

- Визуализация территориального распределения парков и скверов с учетом поли-центричности города.
- Оценка доступности значимых зеленых зон для различных слоев населения (геоанализ пешеходной доступности, транспортной связанности).
- Расчет обеспеченности населения зелеными зонами на основе международных стандартов и демографических данных.

Алматы қаласы цифрландыру басқармасы\АДәндібай А.Е.

– Разработка рекомендаций по размещению новых парков
– Разработка интерактивного карты с поддержкой динамического применения слоев данных влияющих на результат

**Источник данных**: Управление общественных пространств, ГИС-данные, результаты полевых съемок и опросов населения.

**Заказчик**: Управление общественных пространств города Алматы.

### 2.16.1. Визуализация территориального распределения парков и скверов с учетом поли-центричности города

Анализ парков и скверов включал всестороннюю работу по паркам, скверам и зелёным территориям города Алматы. Исследование охватывало сбор и подготовку исходной информации, вычисление показателей экологического состояния, оценку доступности зелёных зон для жителей, анализ обеспеченности районов, интеграцию качественных параметров зеленых насаждений и формирование рекомендаций по обновлению существующих парков и созданию новых.

Использовались данные районных акиматов, открытые источники (OSM), ручной сбор сведений из 2ГИС, спутниковые снимки Google Maps, а также геоданные о деревьях и изображения для расчёта NDVI.

Цель исследования — создание комплексной базы парков, оценка текущего состояния зелёных территорий, выявление дефицитных районов и подготовка обоснованных рекомендаций по строительству новых объектов или модернизации существующей инфраструктуры.

*Подготовка и объединение данных:*

– сбор данных из различных источников о парках, скверах и озелененных территориях;
– очистка данных, проверка геометрии, устранение ошибок;
– объединение данных в единый набор с назначением уникального идентификатора парка;
– интеграция сведений об оснащении (лавочки, урны и т.д.) из OSM;
– добавление информации о зеленых насаждениях;
– подготовка данных для всех последующих аналитических этапов.

Эти данные позволили создать консистентную базу зеленых территорий с единым форматом.

Дополнительно была изучена доступность именно входных зон парков, а не только их внешних границ. Такой подход обеспечивает более точную оценку, так как отражает реальные маршруты передвижения жителей.

На основе уличной сети города был рассчитан доступ населения к паркам:

– создавались буферные зоны по пешей доступности;
– определялась территория покрытия каждого парка;
– оценивалось, какие жилые массивы попадают в зону доступности;
– выявлялись районы, где уровень доступности значительно ниже

Алматы қаласы цифрландыру басқармасы/АДәнлібай А.О.

нормы.

```
from shapely.ops import unary_union
import geopandas as gpd

# Убеждаемся, что все переменные (parks, population_grid, districts, restricted_entrances_utm)
# доступны из вашего Jupyter-окружения.

print('=' * 80)
print('БУФЕРНЫЙ АНАЛИЗ ДОСТУПНОСТИ')
print('=' * 80)

# Убеждаемся что все в UTM проекции для корректных расчетов в метрах
parks_utm = parks.to_crs('EPSG:32643')
population_utm = population_grid.to_crs('EPSG:32643')
districts_utm = districts.to_crs('EPSG:32643')

# ↓↓↓ ИЗМЕНЕНИЕ ЗДЕСЬ ↓↓↓
# Ключевые ID для идентификации ограниченных парков
restricted_park_ids = ['park_00028', 'park_00047']
# ↑↑↑ ИЗМЕНЕНИЕ ЗДЕСЬ ↑↑↑

# Создаем буферы для разных временных интервалов
buffer_distances = {
    '5min': 400,   # 5 минут ходьбы
    '10min': 800,  # 10 минут
    '15min': 1200  # 15 минут
}

buffers_utm = {}

for time_name, distance in buffer_distances.items():
    print(f'\nСоздание буферов {time_name} ({distance}м)...')

    # Создаем списки для буферов
    all_buffers = []

    # 1. Буферы от входов для ограниченных парков
    # Примечание: предполагается, что restricted_entrances_utm содержит только входы
    # для парков из restricted_park_ids.
    entrance_buffers = restricted_entrances_utm.geometry.buffer(distance)
    all_buffers.extend(entrance_buffers.tolist())
    print(f'  Буферы от входов: {len(entrance_buffers)}')

    # 2. Буферы от остальных парков (исключаем ограниченные)
    parks_to_buffer = parks_utm.copy()

    # Имя столбца для ID парков (должно совпадать с фактическим именем в вашем GeoDataFrame)
    id_col = 'park_id' # Используем 'park_id' как наиболее вероятное имя

    # ↓↓↓ ИЗМЕНЕННАЯ ЛОГИКА ФИЛЬТРАЦИИ ↓↓↓

    # Исключаем парки по их ID
    if id_col in parks_to_buffer.columns:
        # Маска для исключения: True, если ID находится в списке ограниченных
        mask_restricted = parks_to_buffer[id_col].isin(restricted_park_ids)

        # Фильтруем: оставляем только те, для которых маска False (НЕ ограниченные)
        parks_to_buffer = parks_to_buffer[~mask_restricted]

        print(f'  Исключено ограниченных парков (по ID): {mask_restricted.sum()}')
    else:
        print(f"  Внимание: Столбец '{id_col}' не найден. Фильтрация не выполнена.")

    # ↑↑↑ ИЗМЕНЕННАЯ ЛОГИКА ФИЛЬТРАЦИИ ↑↑↑

    # Создаем буферы для обычных парков
```

Рисунок – 2.4.2.1. Расчет пешей доступности

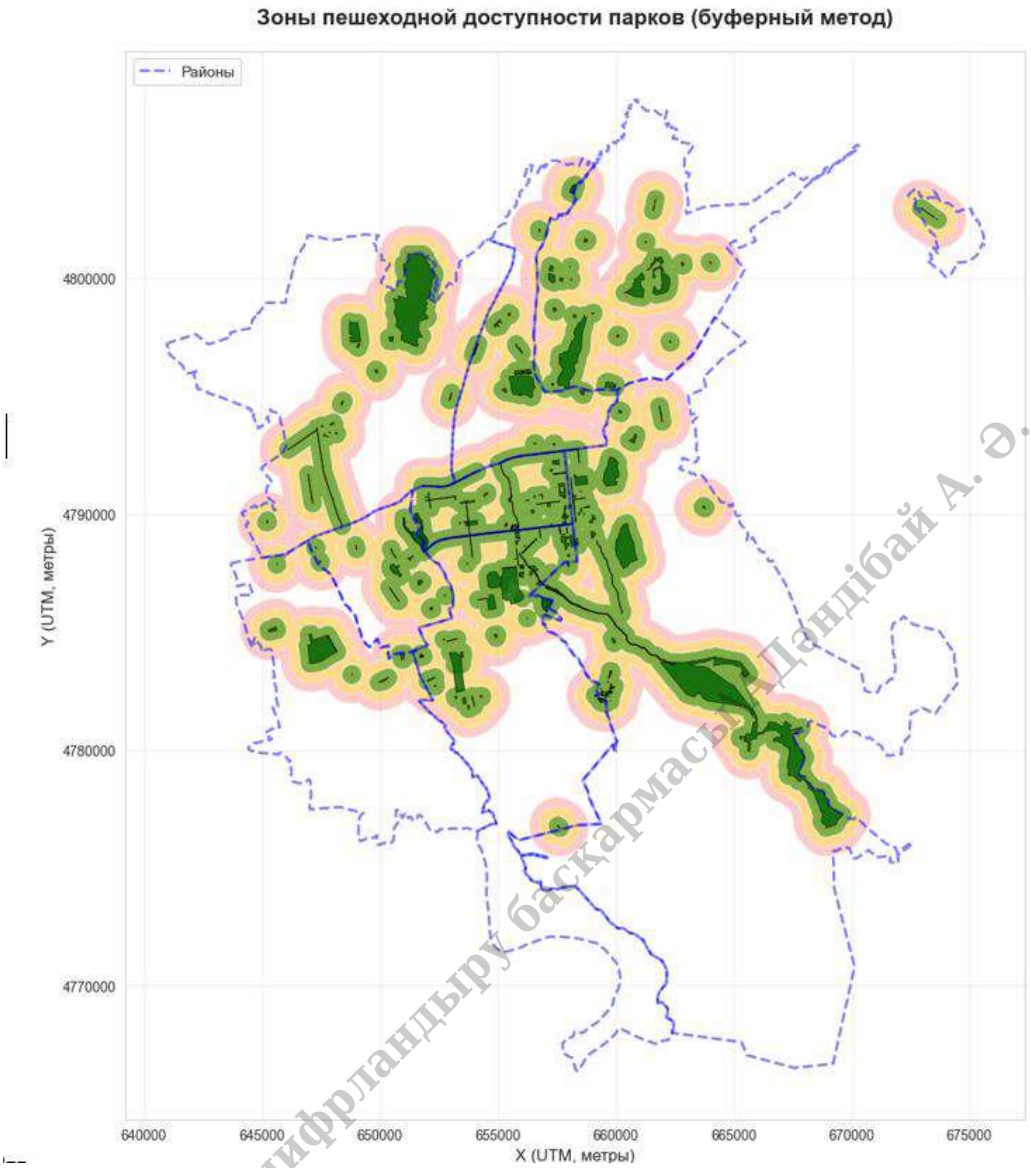



Рисунок – 2.4.2.2. Визуализация зон пешеходной доступности

### 2.16.2. Оценка доступности значимых зеленых зон для различных слоев населения

Был выполнен комплексный анализ ожидаемой обеспеченности населения зелеными зонами, включающий:

- распределение парков по районам;
- расчет обеспеченности по площади;
- оценку обеспеченности по количеству;
- анализ качества озеленения на основе NDVI;
- присвоение интегральных баллов обеспеченности.

Эти данные позволили выявить диспропорции между районами и определить, где потребность в дополнительных зеленых зонах наиболее велика.

Алматы қаласы цифрландыру басқармасы Данібай А. Ә.

```python
print('\n' + '=' * 80)
print('ЗАГРУЗКА ДЕРЕВЬЕВ')
print('=' * 80)

removed_ids_str = ','.join(map(str, removed_statuses))

query = f"""
SELECT
    id,
    breed_id,
    sanitary_id,
    plants_status_id,
    plants_type_id,
    age_group_id,
    disease_id,
    height,
    diameter,
    age,
    ST_AsText(geometry) as geometry_wkt
FROM eco_plants_tree
WHERE plants_type_id = 1
  AND (plants_status_id IS NULL OR plants_status_id NOT IN ({removed_ids_str}))
"""

print('Загрузка деревьев из базы данных...')
print('(это может занять несколько минут - 3+ млн записей)')

trees = pd.read_sql(query, engine)
print(f'\n✓ Загружено {len(trees):,} деревьев')

# Конвертируем WKT в геометрию
print('\nКонвертация геометрии...')
trees['geometry'] = trees['geometry_wkt'].apply(wkt.loads)
trees = gpd.GeoDataFrame(trees, geometry='geometry', crs='EPSG:4326')

# Конвертируем в UTM
trees = trees.to_crs('EPSG:32643')
print('✓ Геометрия конвертирована в EPSG:32643')

# Удаляем ненужный столбец
trees = trees.drop(columns=['geometry_wkt'])

print(f'\nИтого деревьев для анализа: {len(trees):,}')
```

Рисунок – 2.4.2.3. Загрузка данных деревьев с БД

### 2.16.3. Расчет обеспеченности населения зелеными зонами на основе международных стандартов и демографических данных

В рамках анализа зеленых зон были проведены работы:

- проведена интеграция данных о деревьях;
- рассчитана плотность насаждений;
- оценено качество состояния зеленых зон;
- сформированы агрегированные показатели качества по каждому парку.

Качество деревьев является важным фактором оценки экологической значимости зелёной территории.

Для проведения расчетов с базы данных, были загружены данные о зеленых насаждениях на территории парков и скверов.

Алматы қаласы ... Ыдырыс Ыскармасы АДәнсібай А. Ә.

***Расчет NDVI для оценки вегетации***

На основе спутниковых изображений был вычислен индекс NDVI:

- индекс позволяет оценить общее состояние зелёных насаждений;
- использовались снимки на пик сезона;
- NDVI был нормализован и присвоен каждому парку;
- проведена классификация зеленых зон по уровню озеленения.

NDVI является ключевым экологическим параметром, используемым для оценки реального состояния парка, а не только его площади.

```python
def calculate_park_ndvi(park_geometry, start_date, end_date):
    """
    Рассчитывает средний NDVI для полигона парка

    Parameters:
    -----------
    park_geometry : shapely.geometry
        Геометрия парка (Polygon или MultiPolygon)
    start_date : str
        Начальная дата (YYYY-MM-DD)
    end_date : str
        Конечная дата (YYYY-MM-DD)

    Returns:
    --------
    dict : Статистика NDVI (mean, median, min, max, std)
    """
    try:
        # Конвертируем shapely geometry в Earth Engine geometry
        from shapely.geometry import mapping

        # Используем mapping для конвертации любой геометрии
        geom_dict = mapping(park_geometry)

        # Создаем EE геометрию из GeoJSON
        ee_geometry = ee.Geometry(geom_dict)

        # Загружаем коллекцию Sentinel-2 Surface Reflectance
        collection = (
            ee.ImageCollection('COPERNICUS/S2_SR_HARMONIZED')
            .filterBounds(ee_geometry)
            .filterDate(start_date, end_date)
            .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))  # Облачность < 20%
        )

        # Проверяем, есть ли снимки
        count = collection.size().getInfo()
        if count == 0:
            return {
                'mean': None,
                'median': None,
                'min': None,
                'max': None,
                'std': None,
                'count': 0,
                'error': 'No images available'
            }
```

Рисунок – 2.4.2.4. Расчет данных со спутниковых данных Google Satellite

***Рекомендации по модернизации существующих парков***

Проведен анализ инфраструктуры и качества каждого парка для определения необходимости модернизации.

В расчет включались:

- качество озеленения;
- обеспеченность малой архитектурой (лавочки, урны и т.д.);
- состояние доступности;
- функциональные ограничения;
- плотность деревьев и экологическое состояние.

На основе интегральных оценок были выделены:

- парки высокого приоритета для модернизации;
- парки среднего приоритета;
- парки, требующие минимального вмешательства.

Этот блок аналитики дает городской администрации понимание, куда нужно направлять ресурсы в первую очередь.

### 2.16.4. Разработка рекомендаций по размещению новых парков

На основе анализа обеспеченности и доступности определены районы с дефицитом зеленых зон и разработана методология, предоставляющая рекомендации по размещению новых парковых зон.

Разработанные рекомендации включают:

- координаты потенциальных площадок для новых парков;
- расчет зон, наиболее нуждающихся в создании зеленых территорий;
- оценку плотности населения в дефицитных зонах;
- анализ уличной структуры для оптимального размещения;
- определение минимальных размеров необходимых зеленых зон.

Собранные данные помогают градостроителям определять приоритеты в создании новых парков, опираясь на реальные потребности жителей.

Комплексная аналитика парков и скверов включает подготовку данных, многоуровневую экологическую оценку, анализ доступности, оценку обеспеченности населения зелеными территориями и разработку рекомендаций по модернизации и расширению зеленой инфраструктуры города.

В ходе разработки рекомендации, были проделаны следующие работы:

- создана единая очищенная база парков и скверов;
- рассчитаны экологические показатели (NDVI, качество деревьев);
- выполнен анализ зоны покрытия и доступности;
- подготовлены интегральные оценки для всех парков;
- сформированы рекомендации по модернизации существующих парков;
- определены территории для создания новых зеленых зон.

### 2.16.5. Разработка интерактивного карты с поддержкой динамического применения слоев данных влияющих на результат

В рамках выполнения задач по анализу парков и скверов, была разработана

Алматы қаласы цифрландыру басқармасы\АДәндібай А.Ә.

интерактивная веб-платформа для визуализации и анализа состояния парков, скверов и зеленых зон города Алматы.

Система создана на основе современных технологий и предоставляет удобные инструменты для оценки качества городской зеленой инфраструктуры, анализа доступности парков, выявления проблемных территорий и формирования рекомендаций по развитию новых зеленых зон.

Платформа позволяет работать с большим объемом геопространственных данных, проводить детальную аналитику и визуализировать результаты в удобном формате, обеспечивая основу для принятия обоснованных решений городскими службами.

***Особенности реализации***

***Технологический стек***

Для разработки системы применены технологии:

- Next.js 16.0.3 и React 19.2.0;
- TypeScript;
- Leaflet и React-Leaflet для интерактивных карт;
- Turf.js для геопространственного анализа;
- Tailwind CSS для стилизации;
- Recharts для визуализации статистики;
- Radix UI для элементов интерфейса.

Такой стек обеспечивает высокую производительность, адаптивность и расширяемость платформы.

***Структура приложения***

Платформа состоит из трех ключевых разделов.

***1) Главная страница (карта парков)***

Раздел предоставляет интерактивную карту с отображением всех парков и скверов.

Реализована фильтрация по районам, типам объектов и категориям состояния, показ пешеходной и транспортной доступности, отображение входов в парки и анализ обеспеченности населения.

Алматы қаласы цифрландыру басқармасы АДәндібай А. Ә.

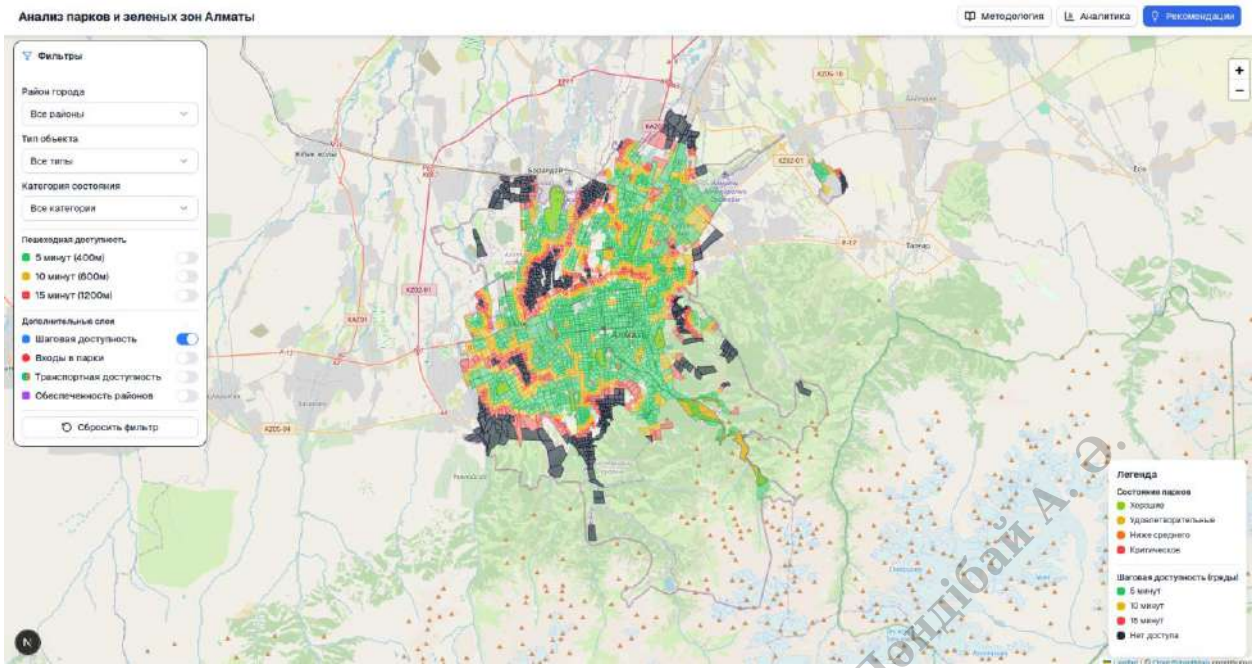



Рисунок – 2.4.4.5. Главная страница со слоем пешей доступности

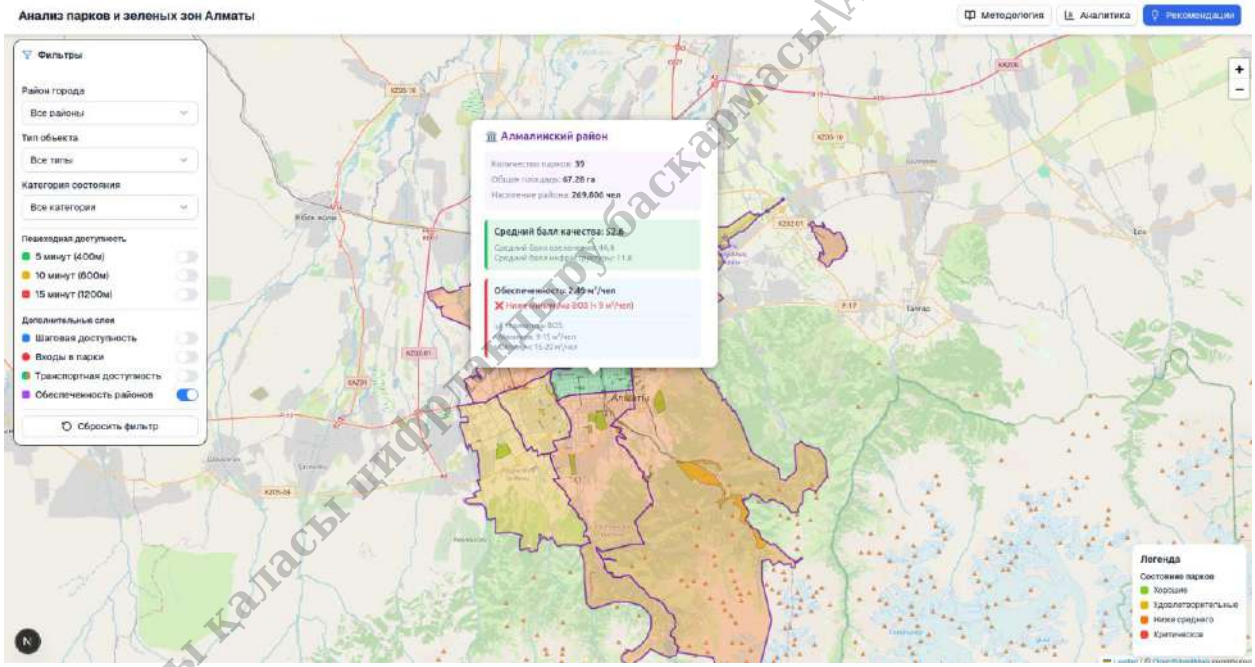



Рисунок – 2.4.2.6. Обеспеченность районов

***2) Страница аналитики***

Содержит статистические графики и сводные показатели: распределение парков по районам, состояние зеленых зон, анализ типов объектов, ключевые KPI и сравнительный анализ районов.

***3) Страница рекомендаций***

Раздел визуализирует приоритеты модернизации и потенциальные локации для создания новых парков. Отображаются буферные зоны, гриды доступности населения, а также методология расчета рекомендаций.

Алматы қаласы цифрлық даму басқармасы\Аденбай А. Э.

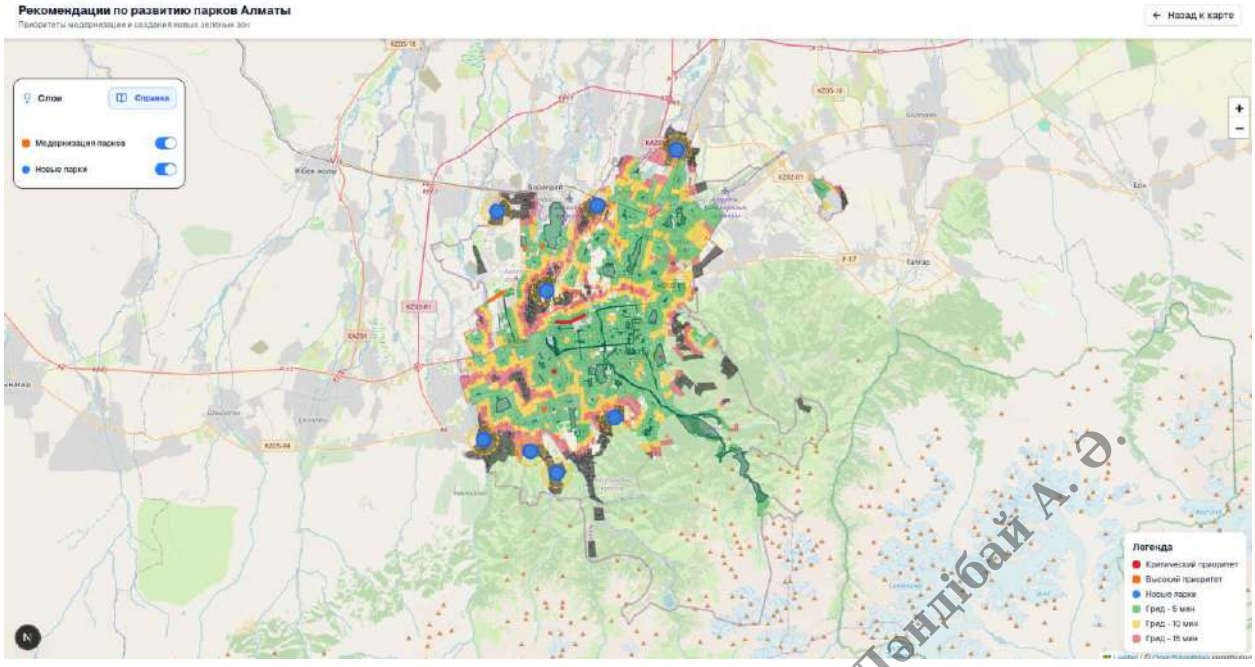



Рисунок – 2.4.2.7. Вкладка рекомендации и модернизации парков

***Картографическая визуализация***

Используется цветовое кодирование состояния парков, интерактивные подсказки, выделение объектов при наведении и многослойная структура карты с удобным управлением слоями.

***Детальная информация о парках***

Для каждого объекта предоставляются данные о характеристиках, инфраструктуре, озеленении, индексах качества, обслуживаемом населении и истории ремонтов.

***Визуализация рекомендаций***

Приоритеты модернизации отображаются цветами, зонами буферов и маркерами новых парков. Гриды доступности показывают, насколько население обеспечено зелеными зонами.

Интерфейс разработан с учетом удобства работы:

- верхняя панель навигации по разделам;
- панель фильтров слева с возможностью управления слоями;
- модальные окна методологии и справки;
- адаптивная вёрстка и поддержка темной темы.

В результате работы создана полноценная веб-платформа для анализа парков и зеленых зон города Алматы. Система позволяет:

- оценивать состояние каждого парка по множеству критериев;
- анализировать обеспеченность районов и доступность парков для населения;
- выявлять территории, требующие модернизации;
- определять оптимальные места для новых парков.

Алматы қаласы цифрландыру басқармасы\Аңдібай А. Ә.