

Разработка чат-бота в поддержку абитуриента ВятГУ

Научный руководитель доцент
кафедры ПМИ Разова Елена
Владимировна

Выполнил студент 4 курса
Ощепков Дмитрий Олегович

Актуальность

- **Актуальность темы** исследования обусловлена необходимостью повышения качества и доступности информационной поддержки абитуриентов ВятГУ.

Цель

- **Цель работы** – разработка Telegram-бота для предоставления справочной информации и рекомендаций абитуриентам ВятГУ на основе данных, размещенных на официальных источниках ВятГУ.

Задачи

Для реализации поставленной цели необходимо решить следующие **задачи**:

- собрать информацию с сайта для абитуриентов ВятГУ;
- систематизировать полученную информацию для эффективного доступа к ней;
- разработать архитектуру Telegram-бота с удобным пользовательским интерфейсом;
- реализовать рекомендательную систему на основе описания направлений;
- провести тестирование работоспособности и удобства использования чат-бота.

Сильные и слабые стороны существующих аналогов

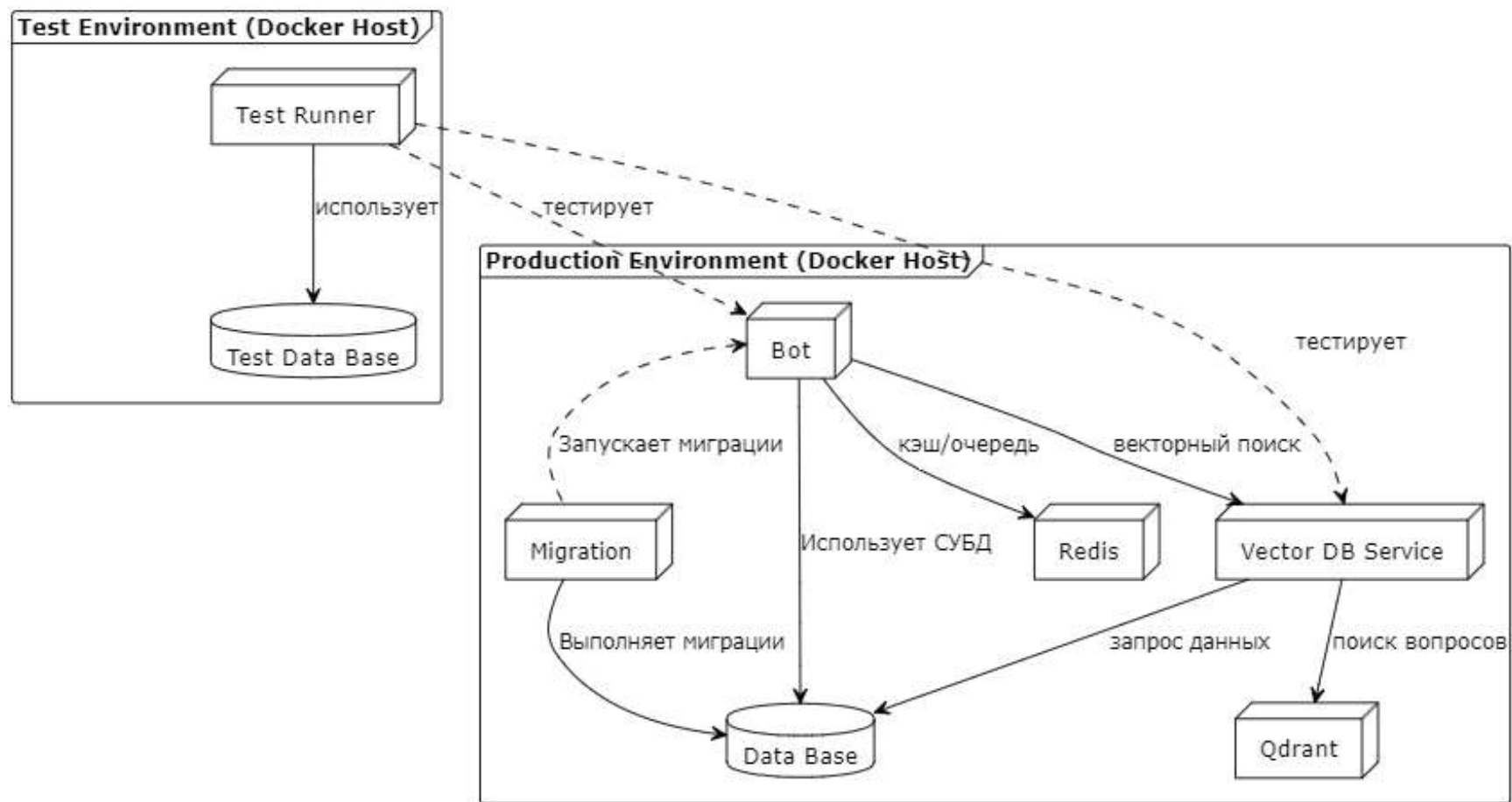
- **Сильные стороны:**

- Возможность подписки на обновления
- Поддержка нескольких языков
- Доступ к официальным источникам
- Свободный ввод и подключение оператора
- Структурированная информация

- **Слабые стороны:**

- Отсутствие рекомендаций по программам
- Переключение между клавиатурами
- Избыточные визуальные элементы

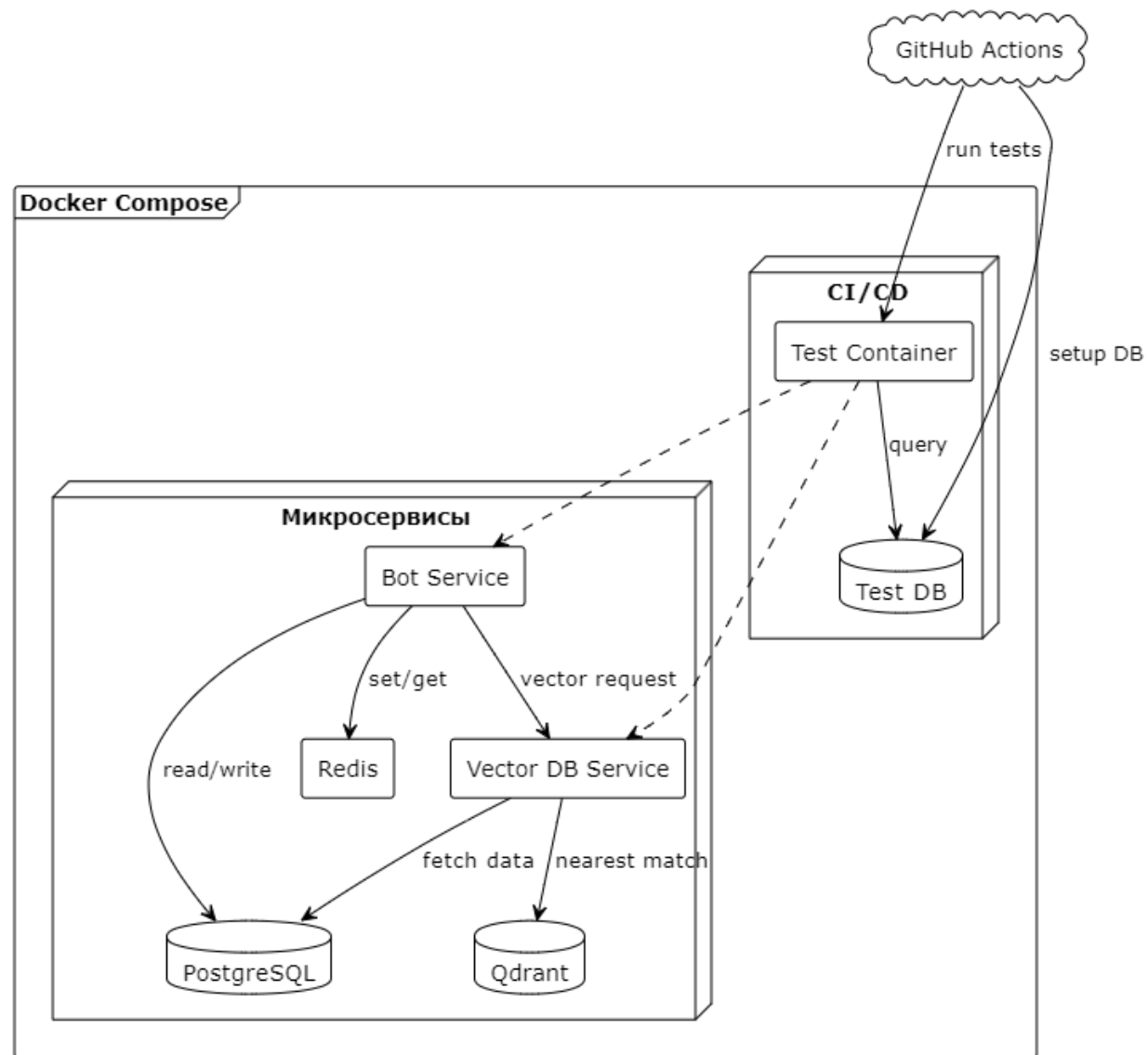
Компоненты и архитектура



Процесс развертывания

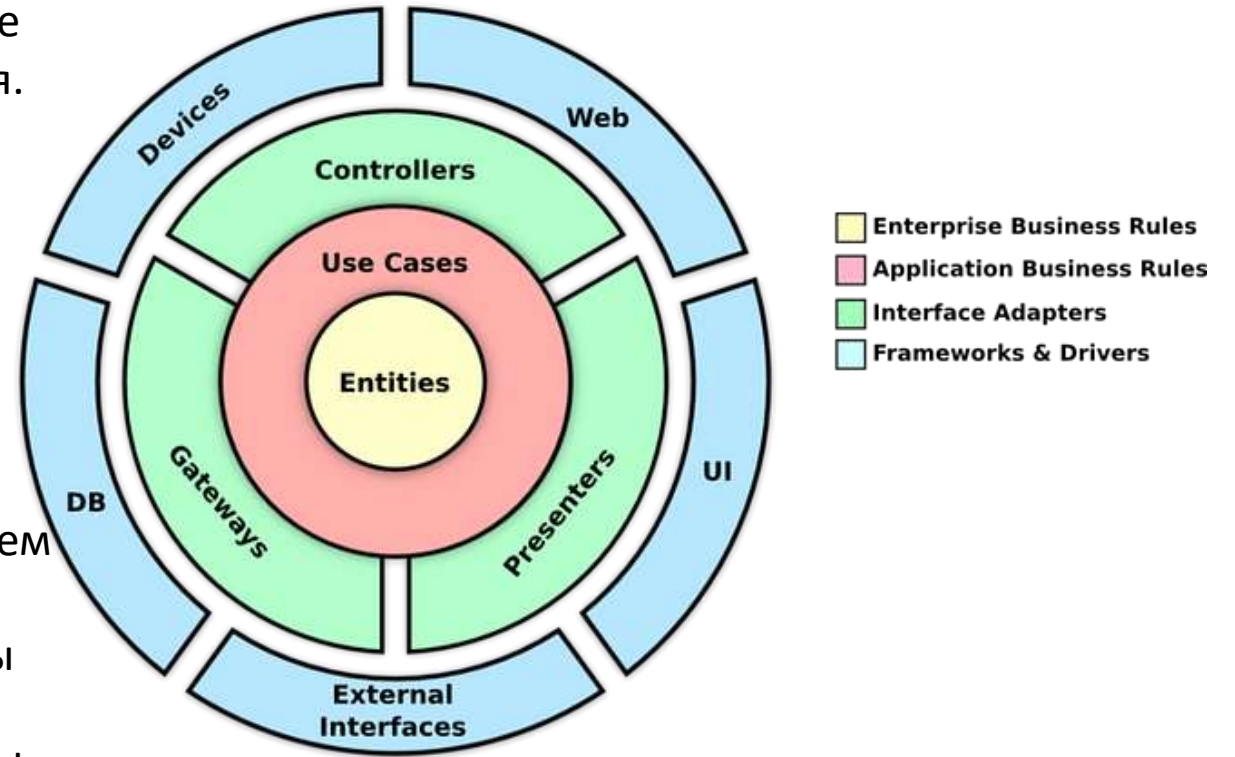
- Вначале разворачиваются независимые контейнеры: Data Base, Qdarant, Redis, Test Data Base;
- после развертывании контейнер Migration запускает миграционные скрипты и выполняет их над Data Base;
- одновременно с этим Test Runner запускает тесты;
- запускается Vector DB Service и сразу же обращается к базе данных для обновления Qdaran;
- в конце запускается Bot, который так или иначе используется все компоненты в Prodaction окружении.

GitHub Actions

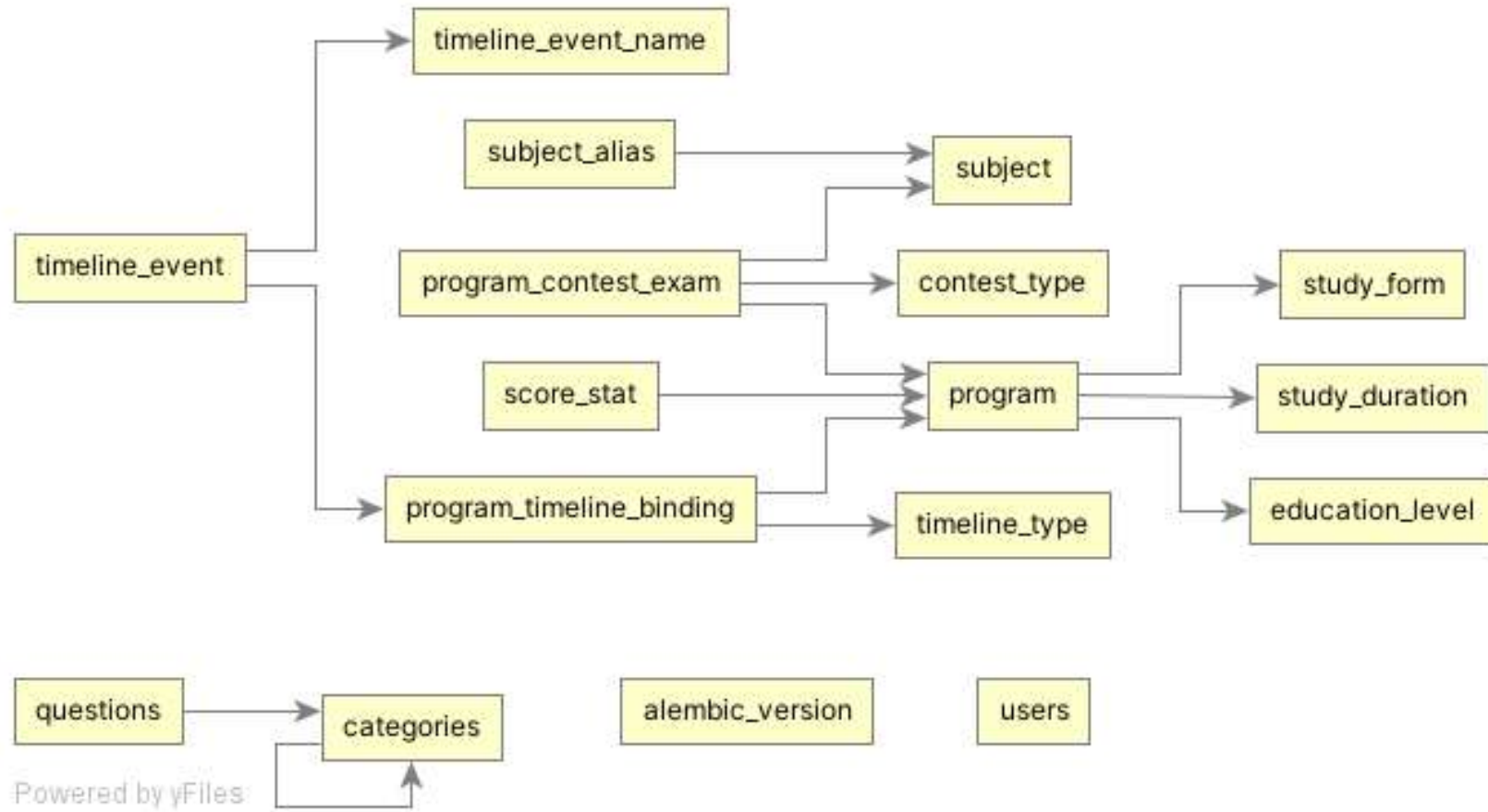


Архитектура основного сервиса – бота

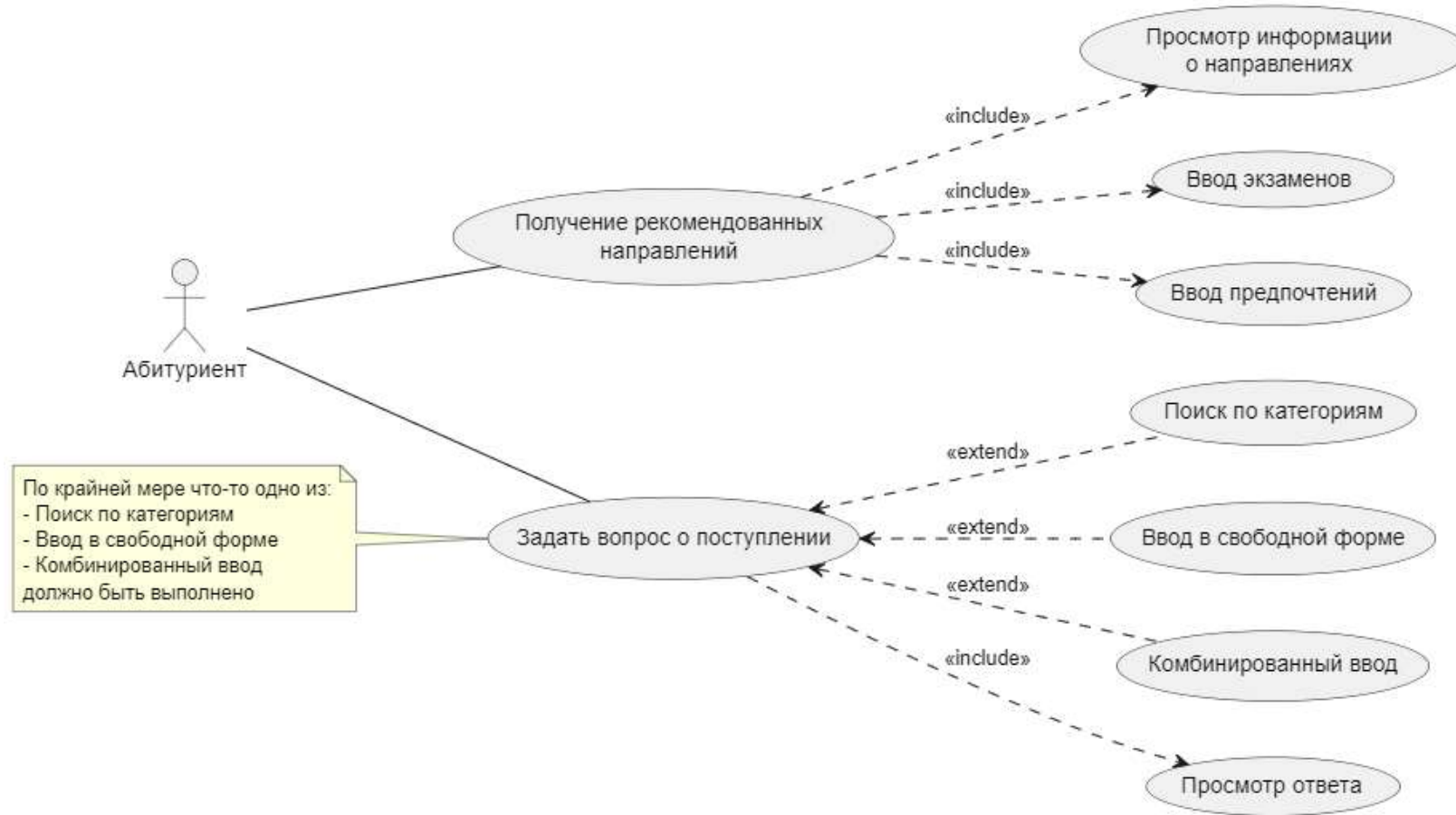
- Домен (на рисунке Enterprise Business Rules). Он содержит бизнес-объекты и логику, которая не зависит от внешних систем и должна меняться.
- Слой приложения (на рисунке Application Business Rules). В нем описываются бизнес-правила и сценарии использования. Он использует интерфейсы, а не конкретные реализации.
- Слой представления (на рисунке Interface Adapters). Отвечает за взаимодействие с пользователем через интерфейс Telegram. В нем расположены интерфейсы репозитория для взаимодействия с базой данных и интерфейсы других сервисов.
- Инфраструктурный слой (на рисунке Frameworks & Drivers). Здесь расположены конкретные реализации для связи с внешними сервисами.



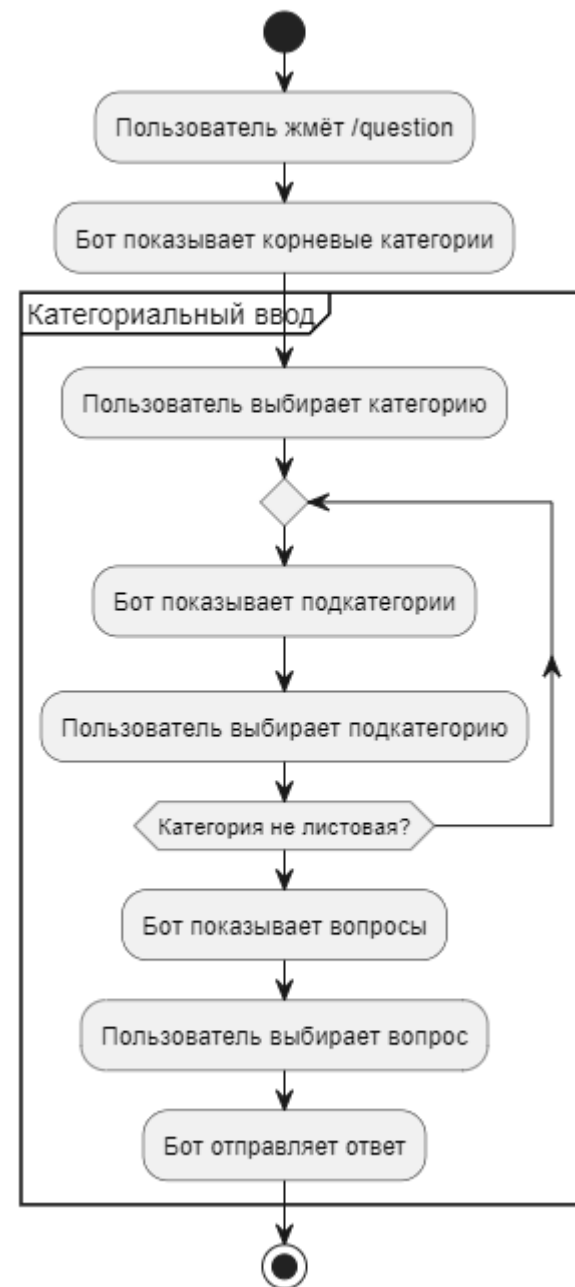
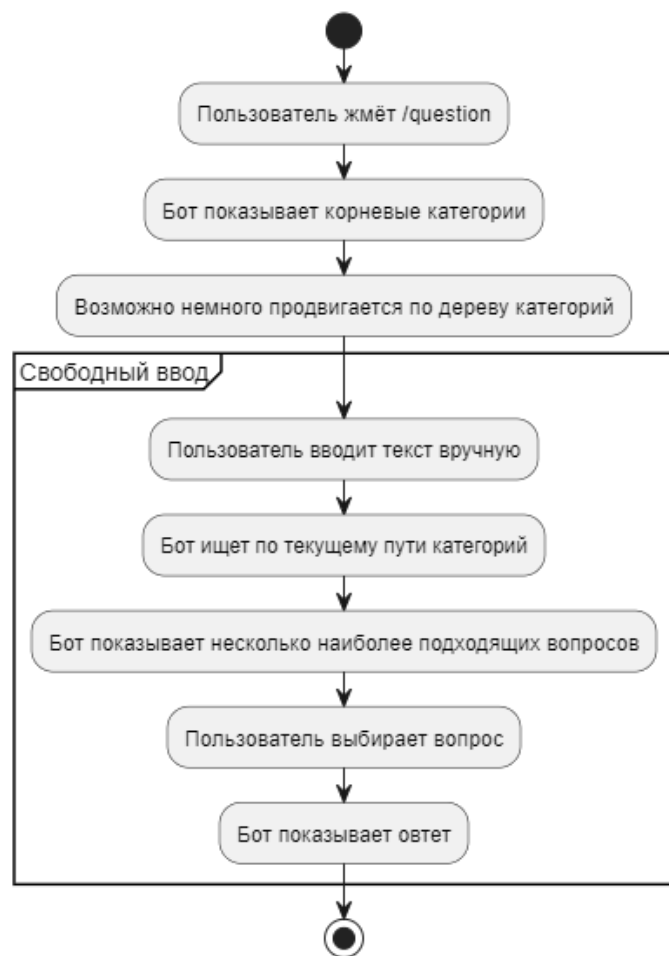
Структура PostgreSQL



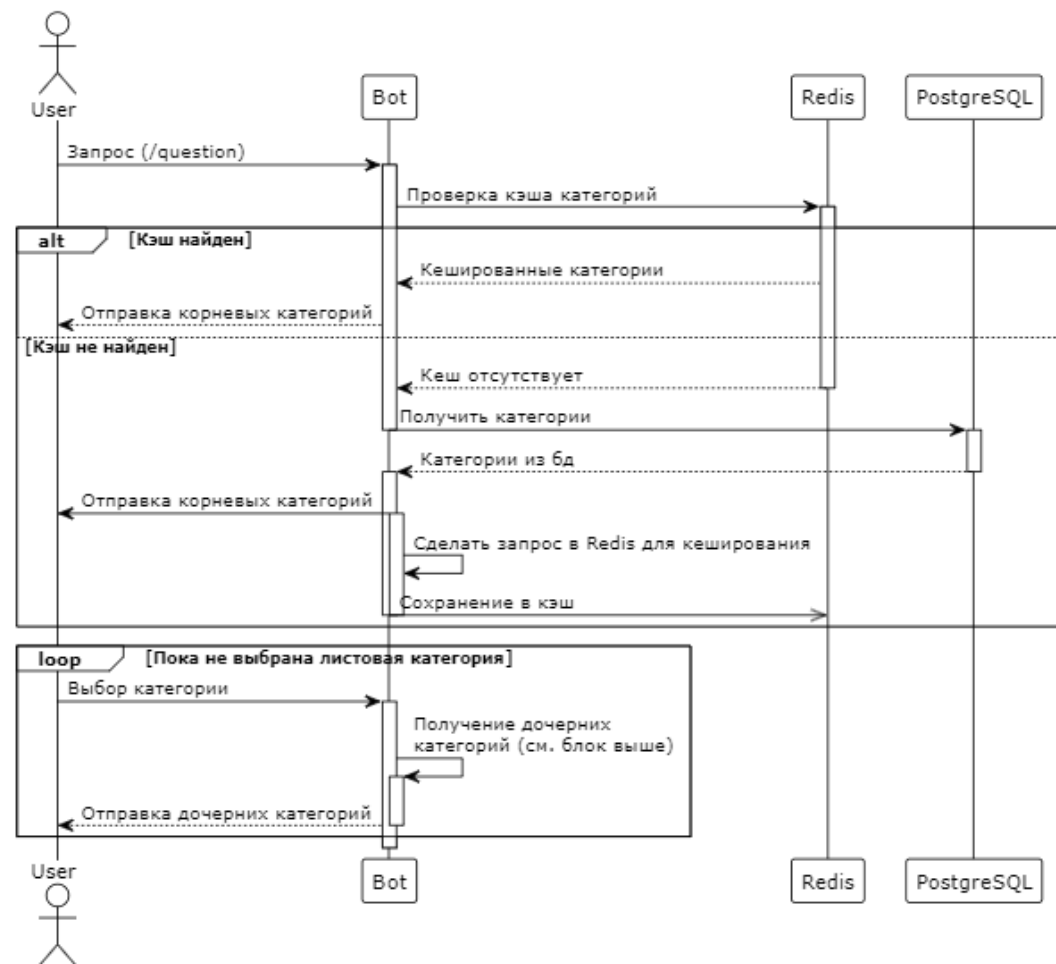
Сценарии взаимодействия



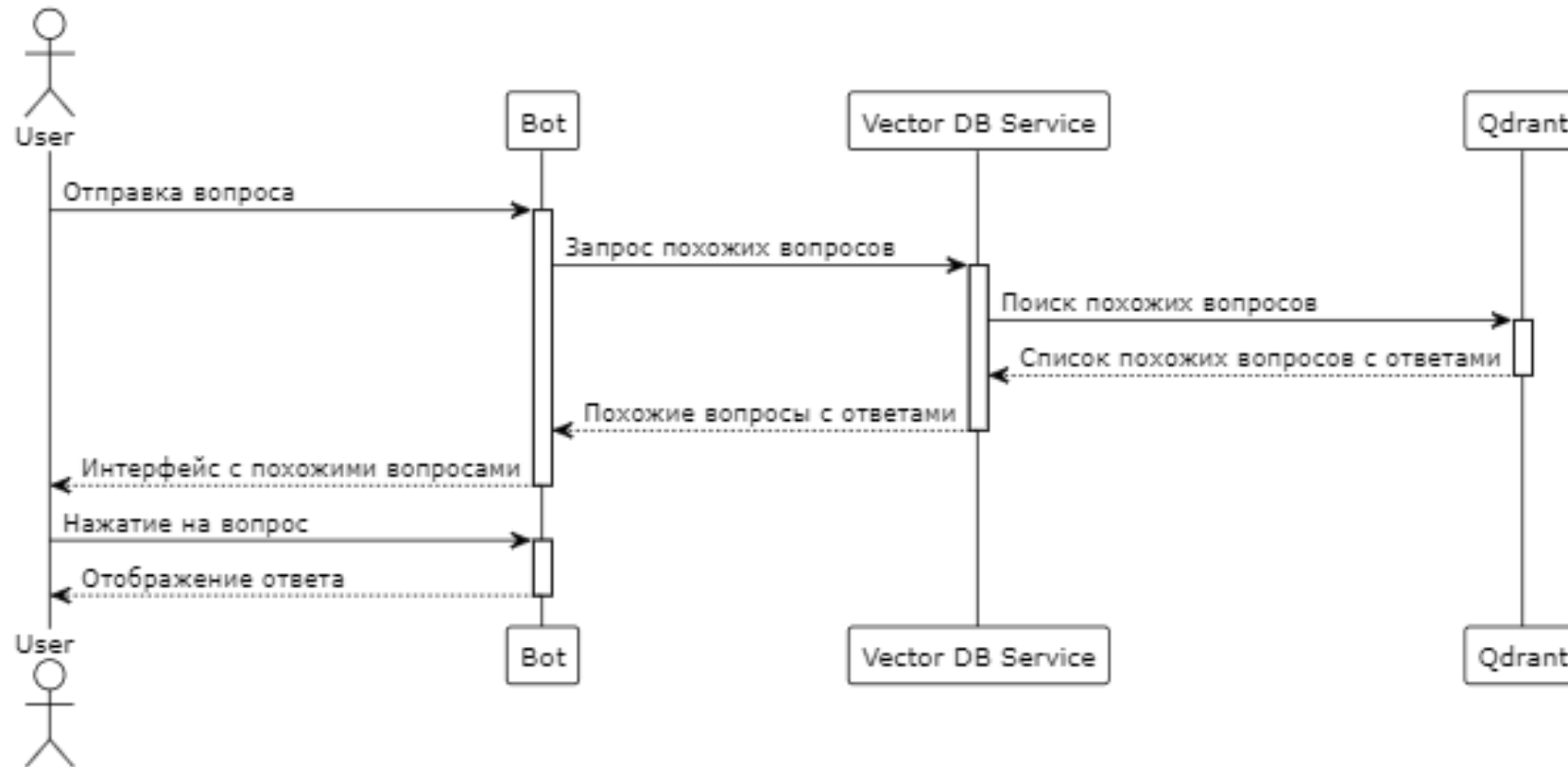
Вопрос о поступлении



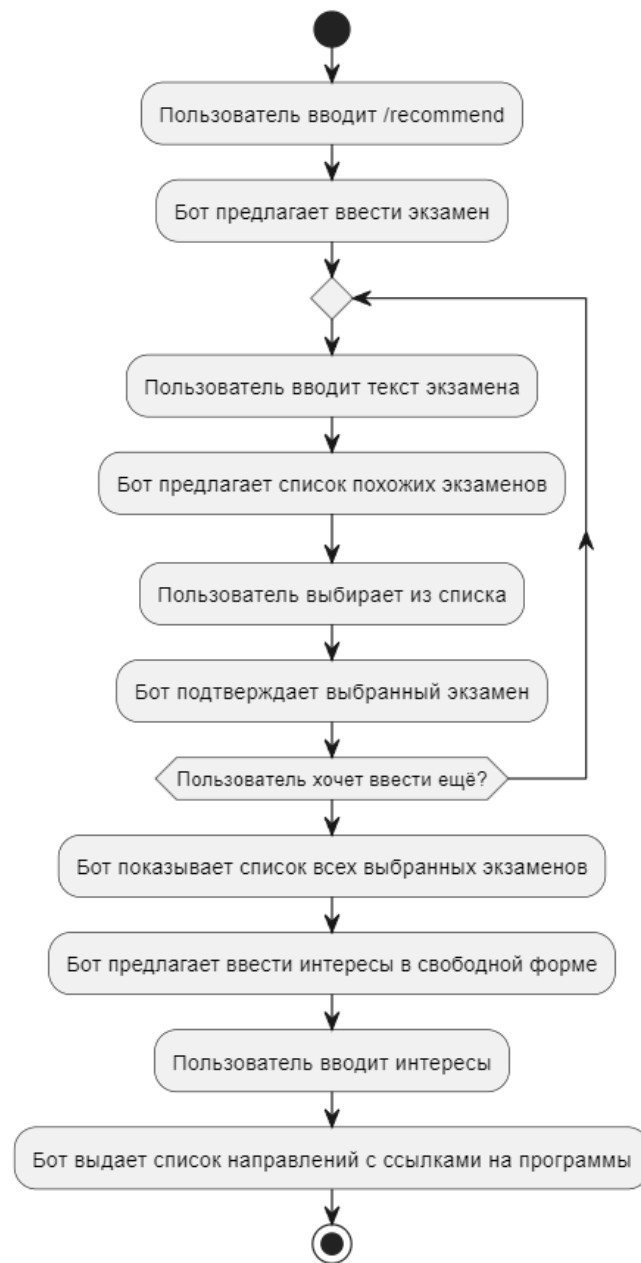
Поиск по категориям



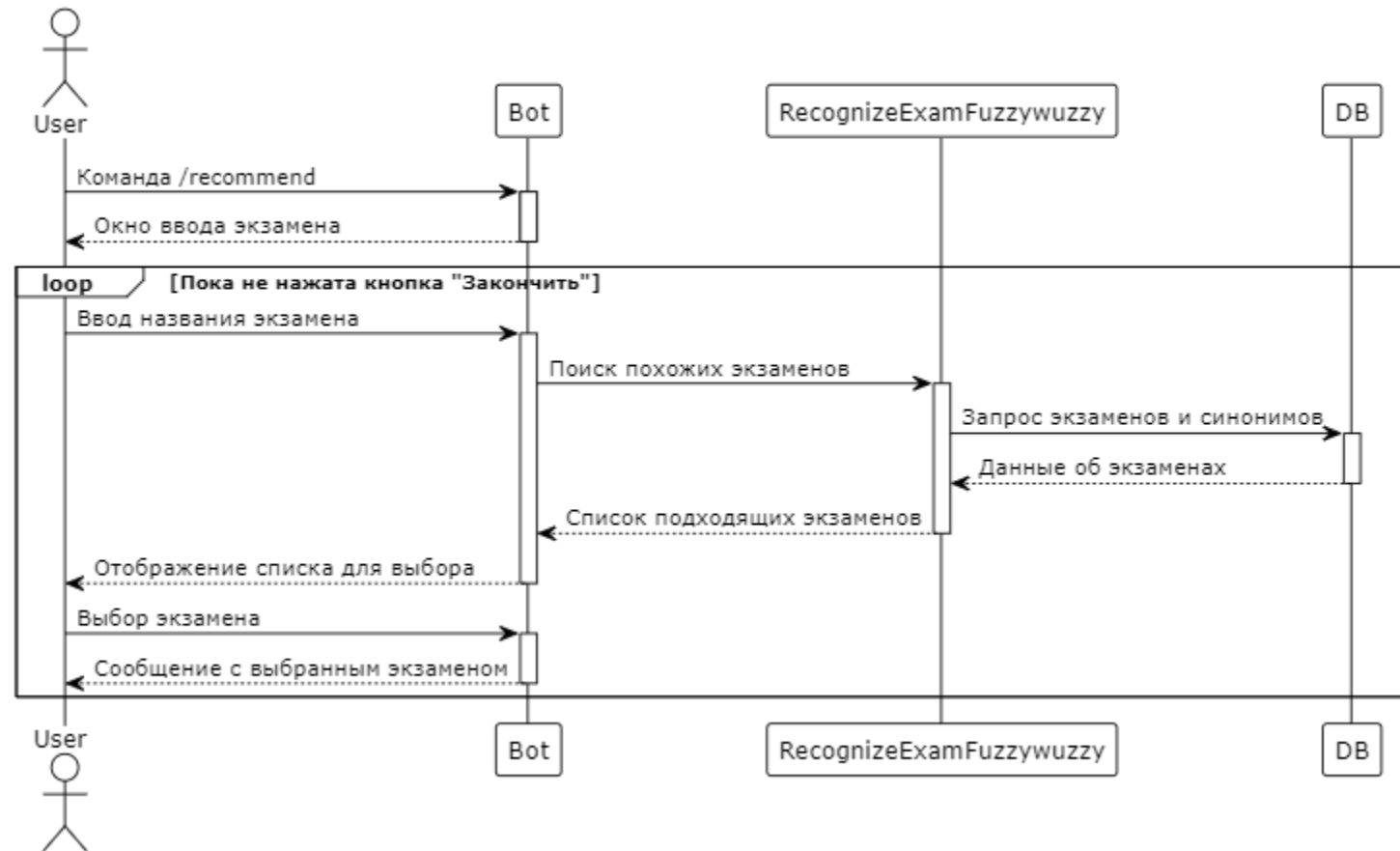
СВОБОДНЫЙ ВВОД



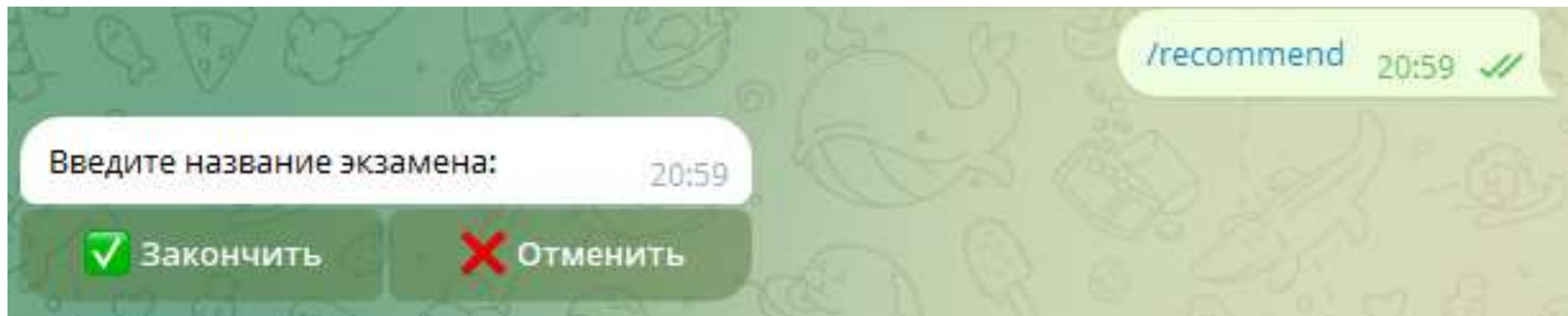
Получить рекомендации



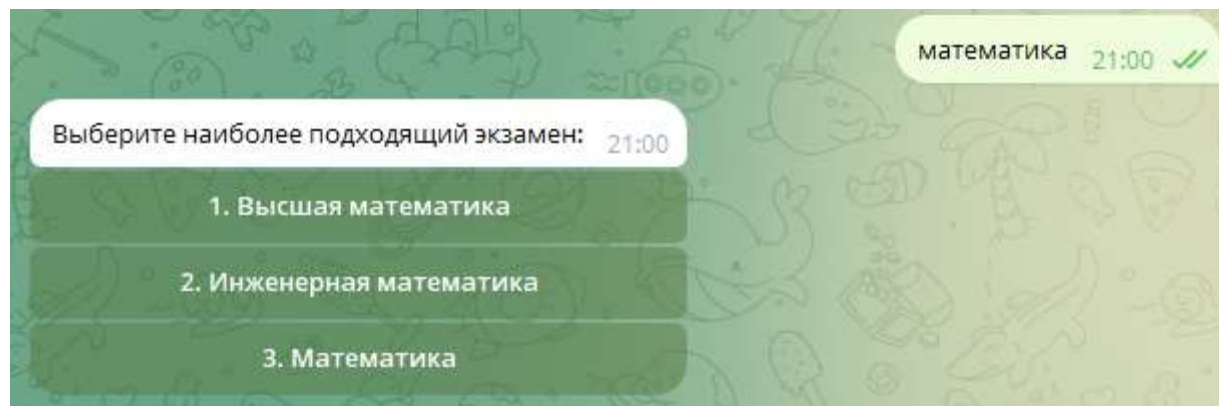
Взаимодействие с сервисом исправления опечаток в предметах



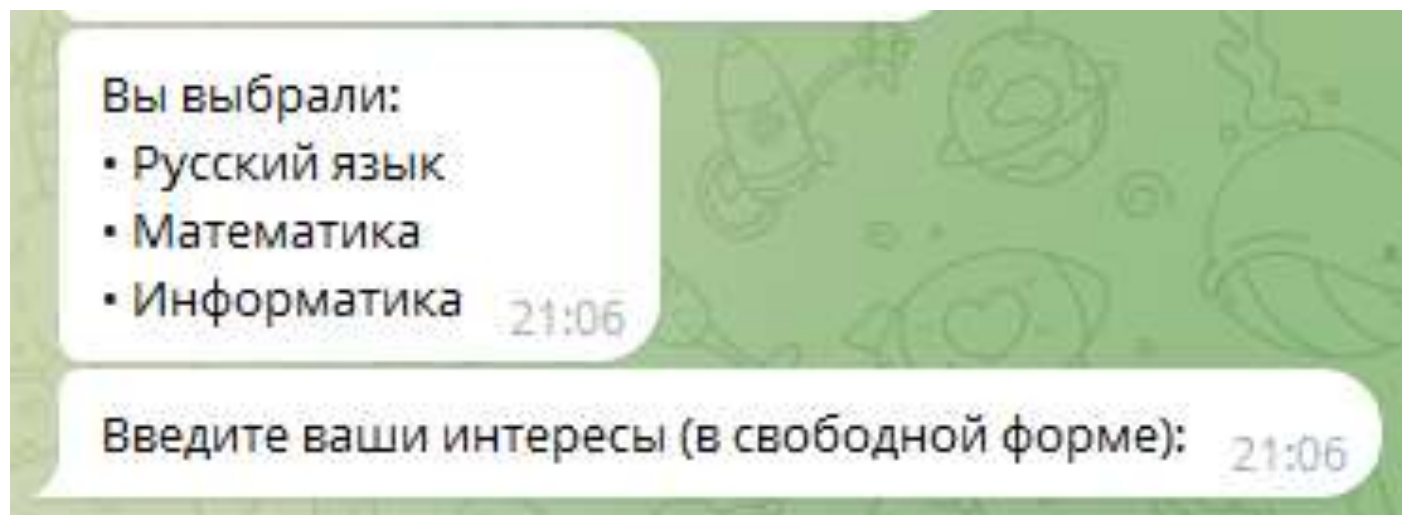
Демонстрация интерфейса



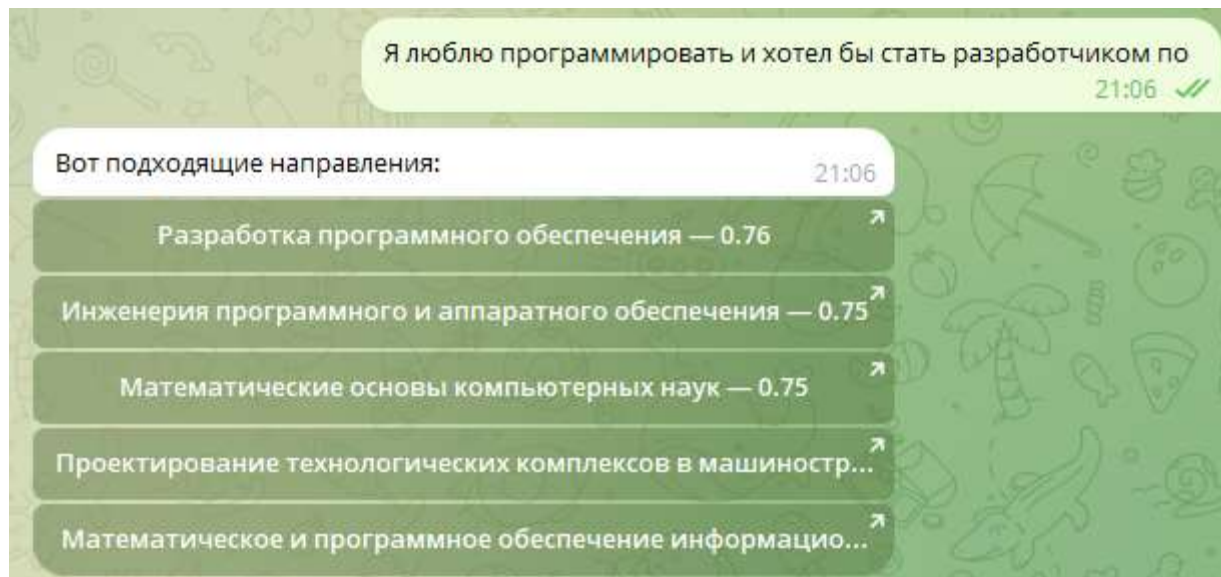
Демонстрация интерфейса



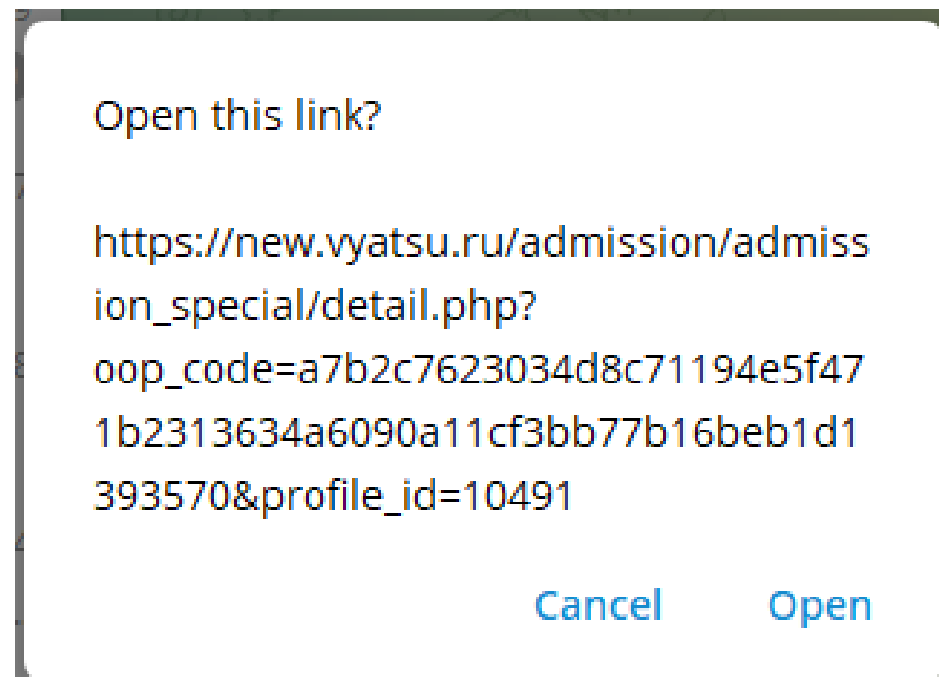
Демонстрация интерфейса



Демонстрация интерфейса



Демонстрация интерфейса



Перспективы

- В будущем планируется реализовать:
 - механизм подписки на уведомления о важных событиях;
 - аналитику и учетность для администраторов системы;
 - автоматическое обновление данных раз в определенное время (на данный момент процесс полуавтоматический);
 - web-интерфейс на базе ядра приложения, реализованного в рамках чистой архитектуры.

Заключение

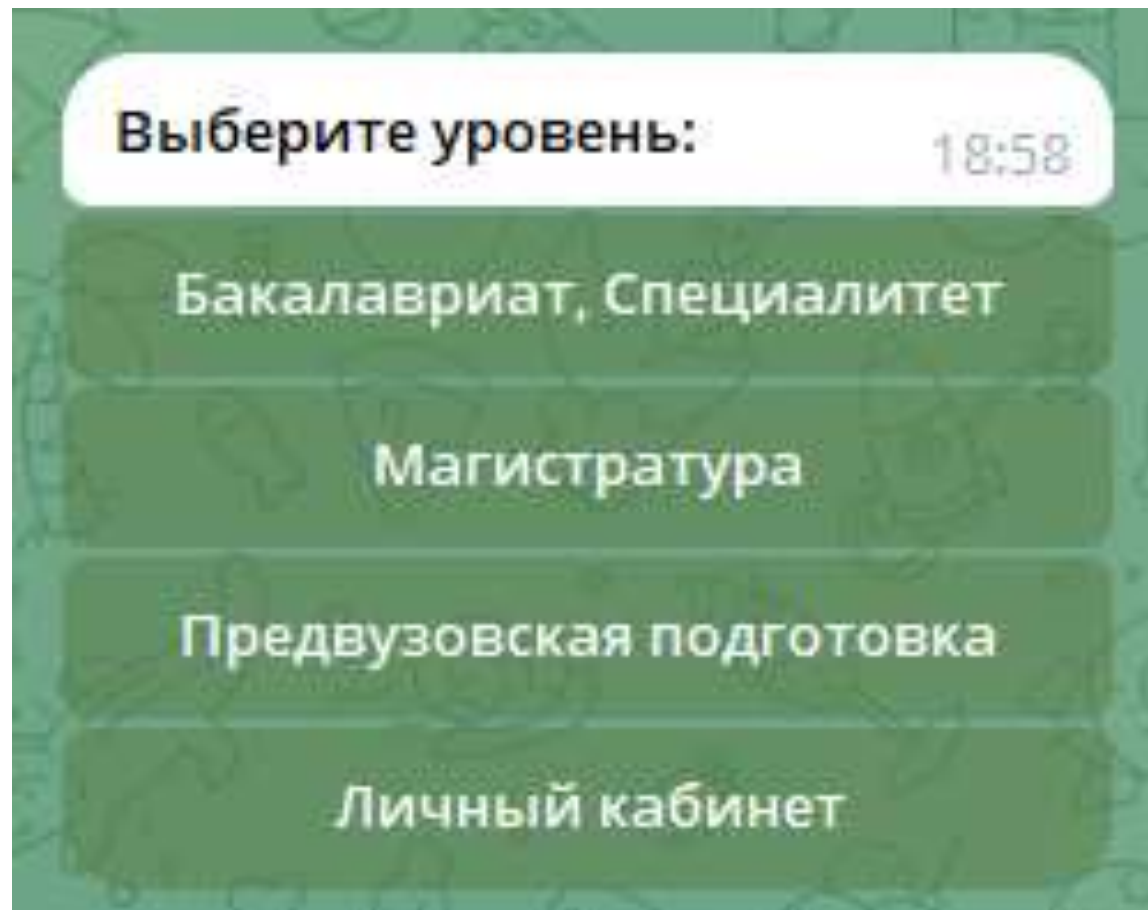
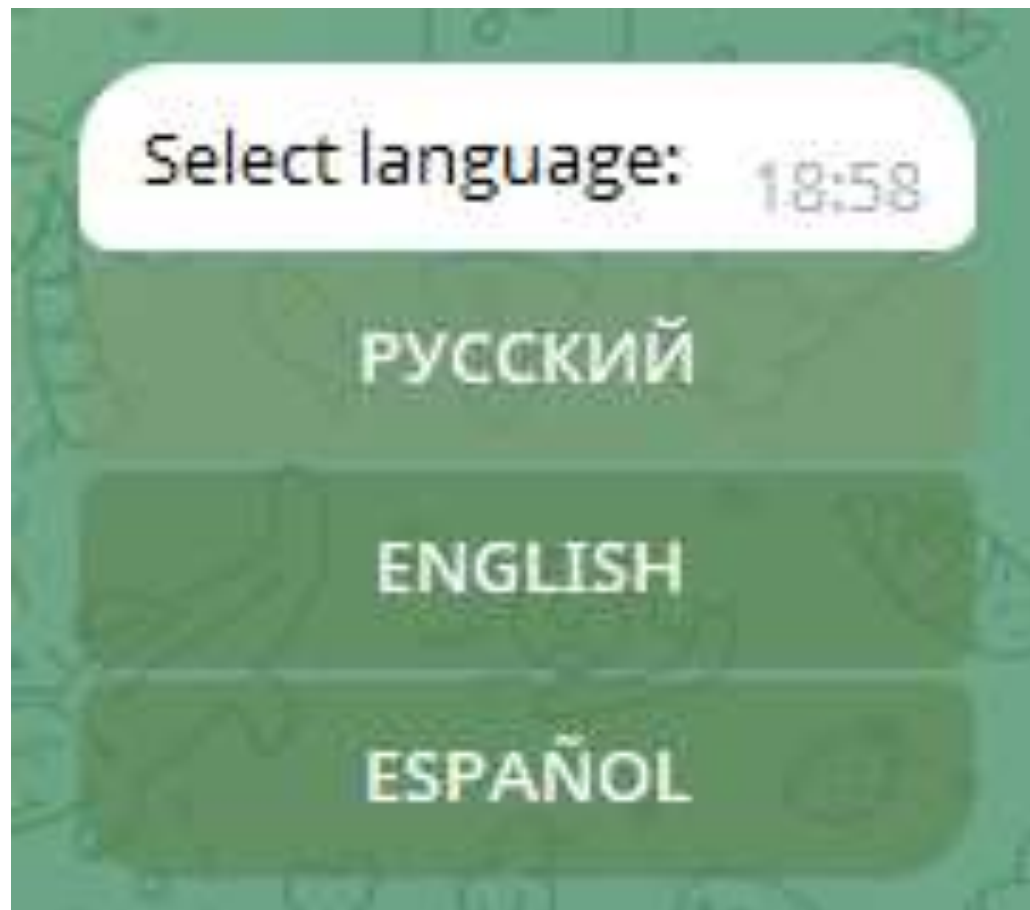
- В рамках данной работы был спроектирован и реализован Telegram-бот, предназначенный для информационной поддержки абитуриентов
- Архитектура приложения обеспечивает высокую модульность, тестируемость и гибкость
- Проект полностью контейнеризирован с использованием Docker
- Система поддерживает как ручной, так и интеллектуальный ввод

Спасибо за внимание!

Аналоги. Admissions KFU

- Официальный бот для иностранных абитуриентов Казанского (Приволжского) федерального университета (КФУ).
- Преимущества:
 - поддержка разных языков;
 - реализует подписки на события образовательных программ;
 - содержит часто задаваемые вопросы;
 - содержит информацию о образовательных программах.
- Недостатки:
 - нет возможности задать вопрос в свободной форме.

Аналоги. Admissions KFU



Аналоги. Admissions KFU

Как поступить на контракт?

Для поступления в рамках отдельного конкурса для иностранных граждан на контрактную форму обучения вам необходимо сделать следующее:

Шаг 1. С 1 марта и по 15 августа 2021г. необходимо подать заявление, приложив все требуемые документы, через социально образовательную сеть "[Буду студентом!](#)".

Шаг 2. Успешно сдать вступительные испытания.

Шаг 3. Заключить договор и оплатить обучение.

Более подробная информация доступна по [ссылке](#).

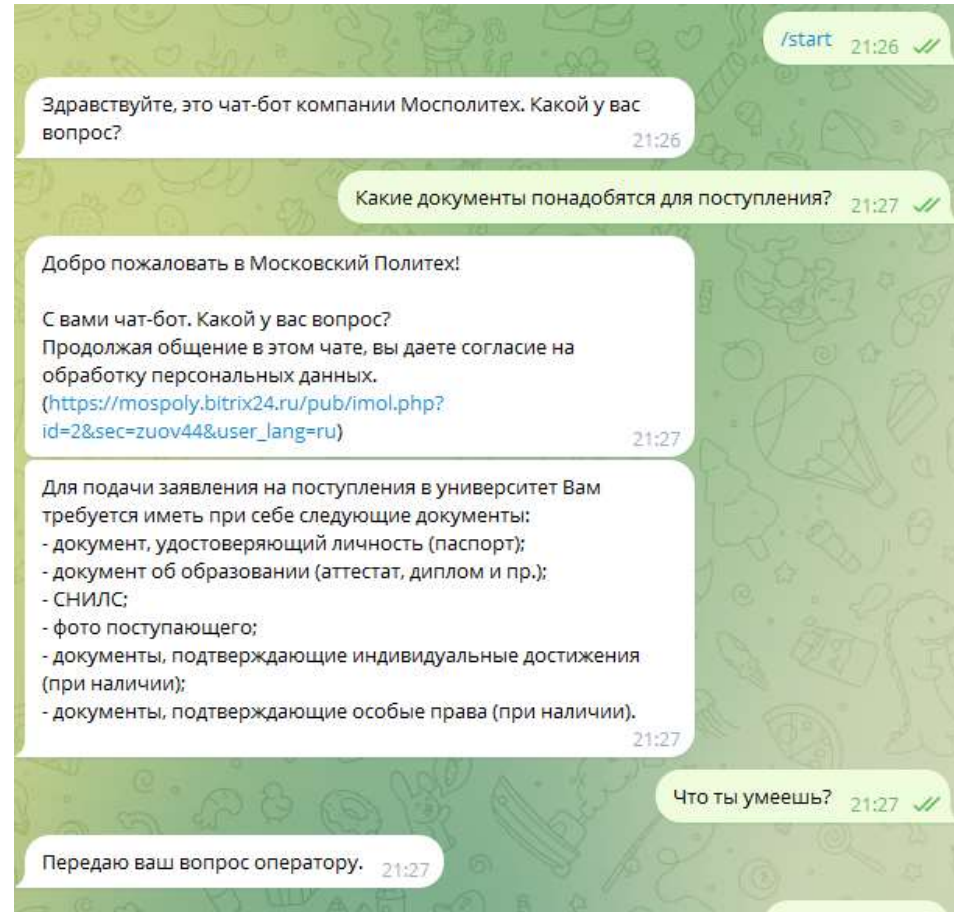
18:58

Назад

Аналоги. Московский политех

- Бот Московского политехнического университета. Позволяет задавать вопросы в свободной форме. Если не находит ответ в базе направляет на оператора.
- Не ясно, что в целом умеет бот. По всей видимости, он предусматривает только задание вопроса в свободной форме.

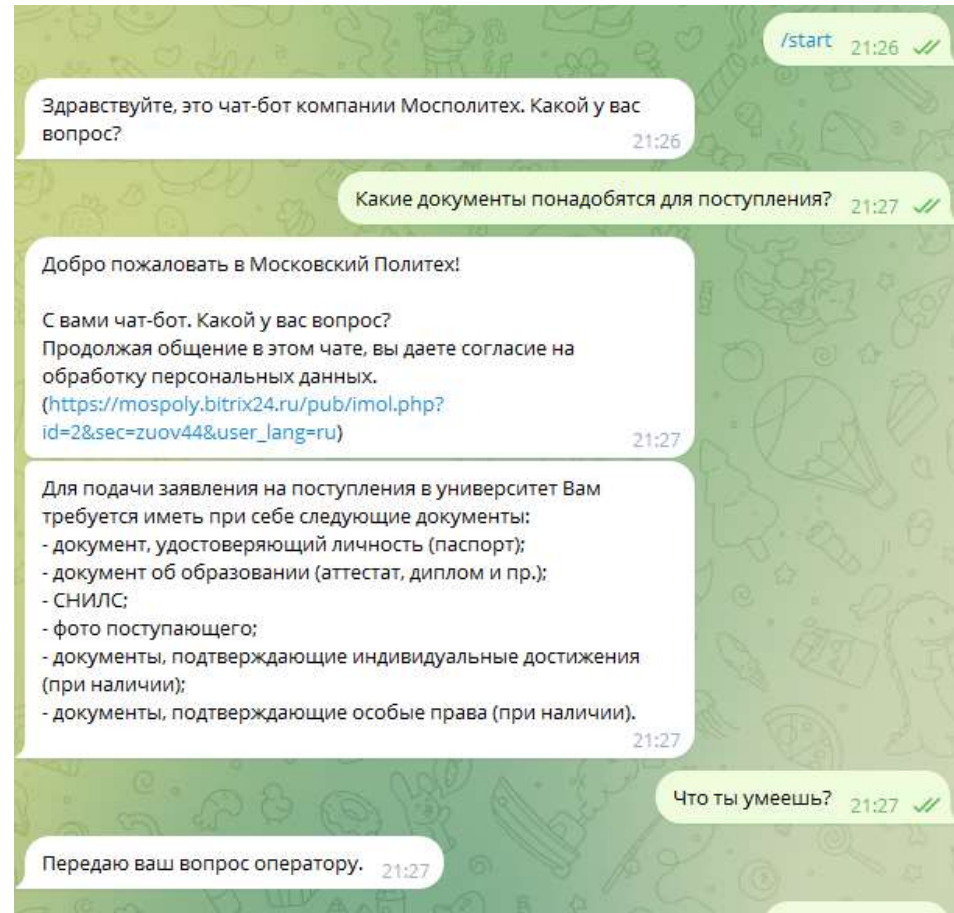
Аналоги. Московский политех



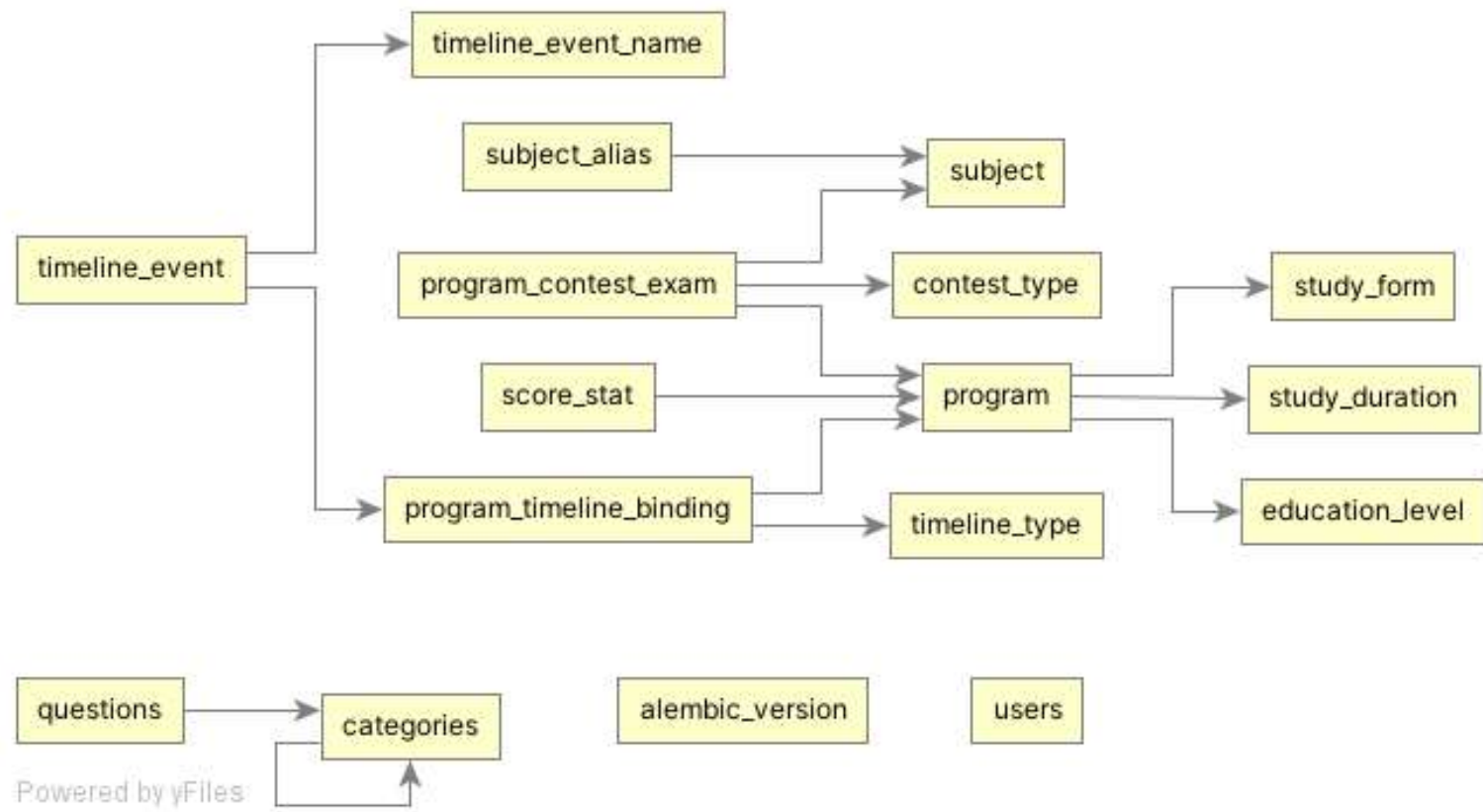
Аналоги. Хочу в Политех

- Бот университета СПбГМТУ (Санкт-Петербургский государственный морской технический университет)
- Преимущества:
 - позволяет открывать нормативные документы не выходя из телеграмма;
 - содержит информацию для иностранных студентов;
 - предоставляет сведения о приёмной кампании;
 - возможность связаться с оператором;
 - к большинству ответов приложена ссылка на актуальный источник.
- Минусы:
 - сочетание Inline и обычной клавиатуры не очень удачно;
 - отсутствует раздел с рекомендациями направлений.

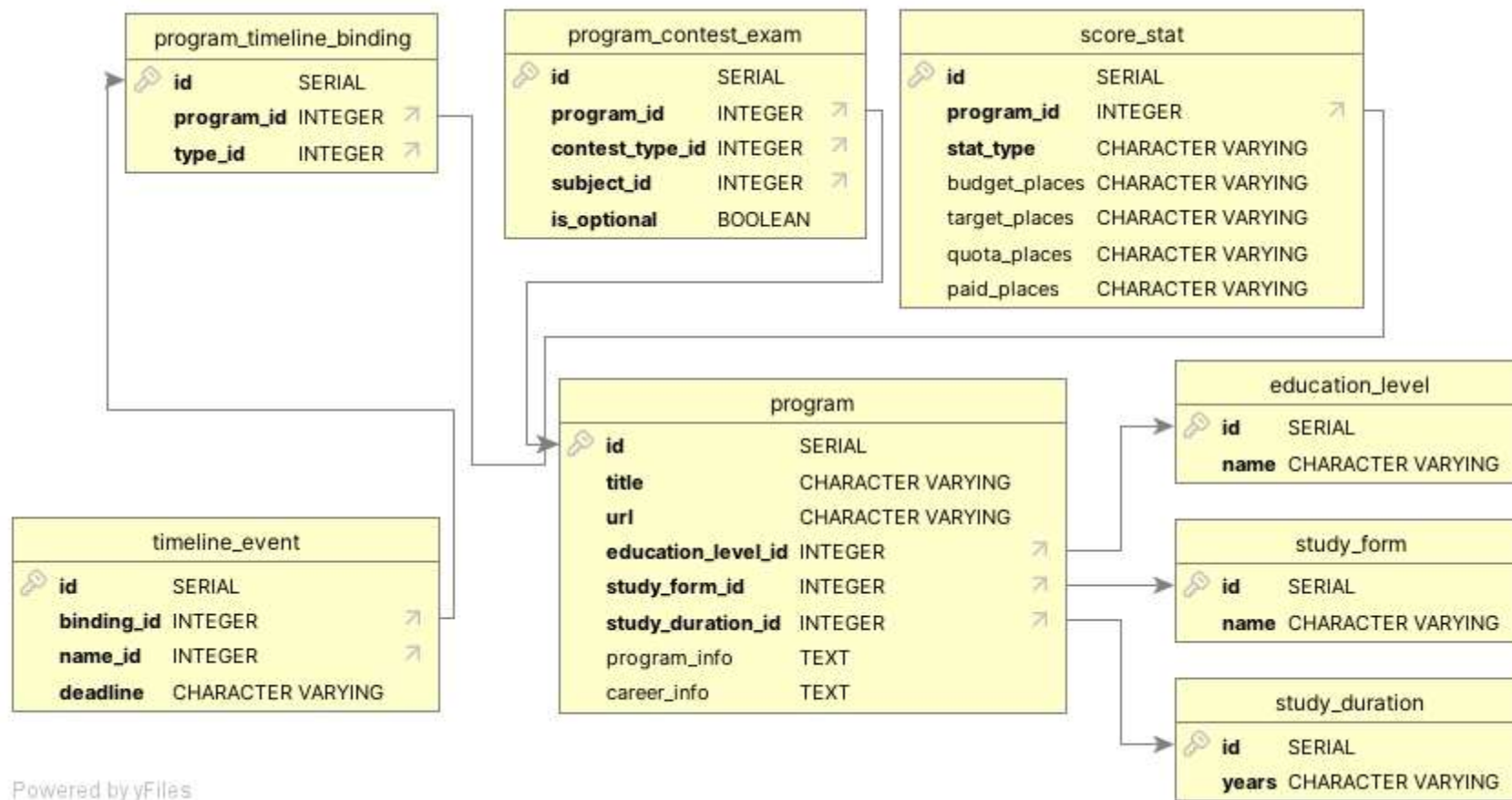
Аналоги. Хочу в Политех



База данных

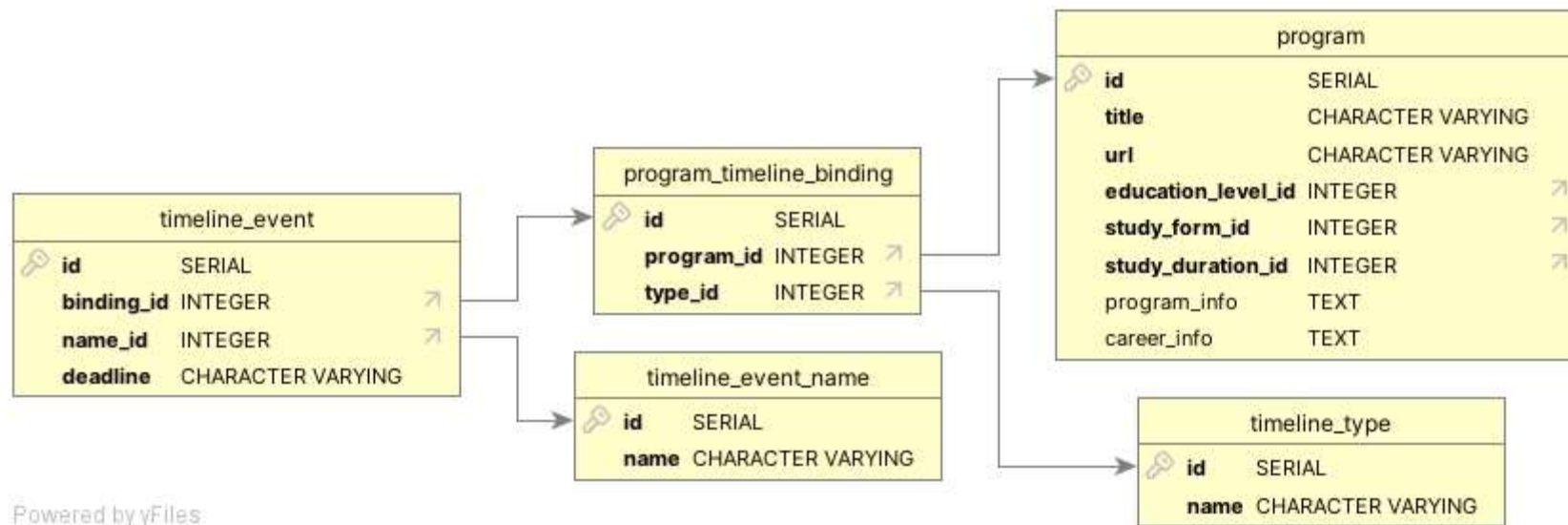


База данных



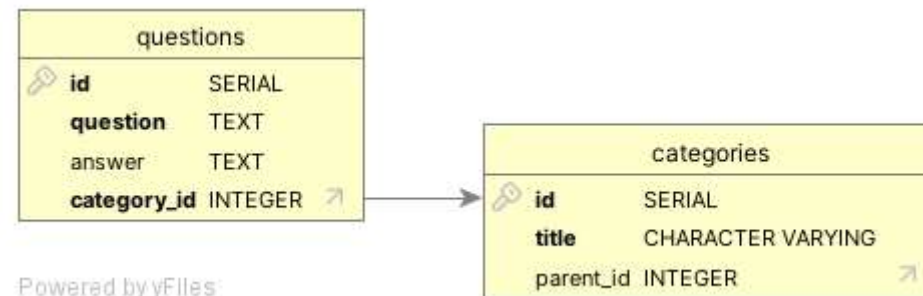
Powered by yFiles

База данных

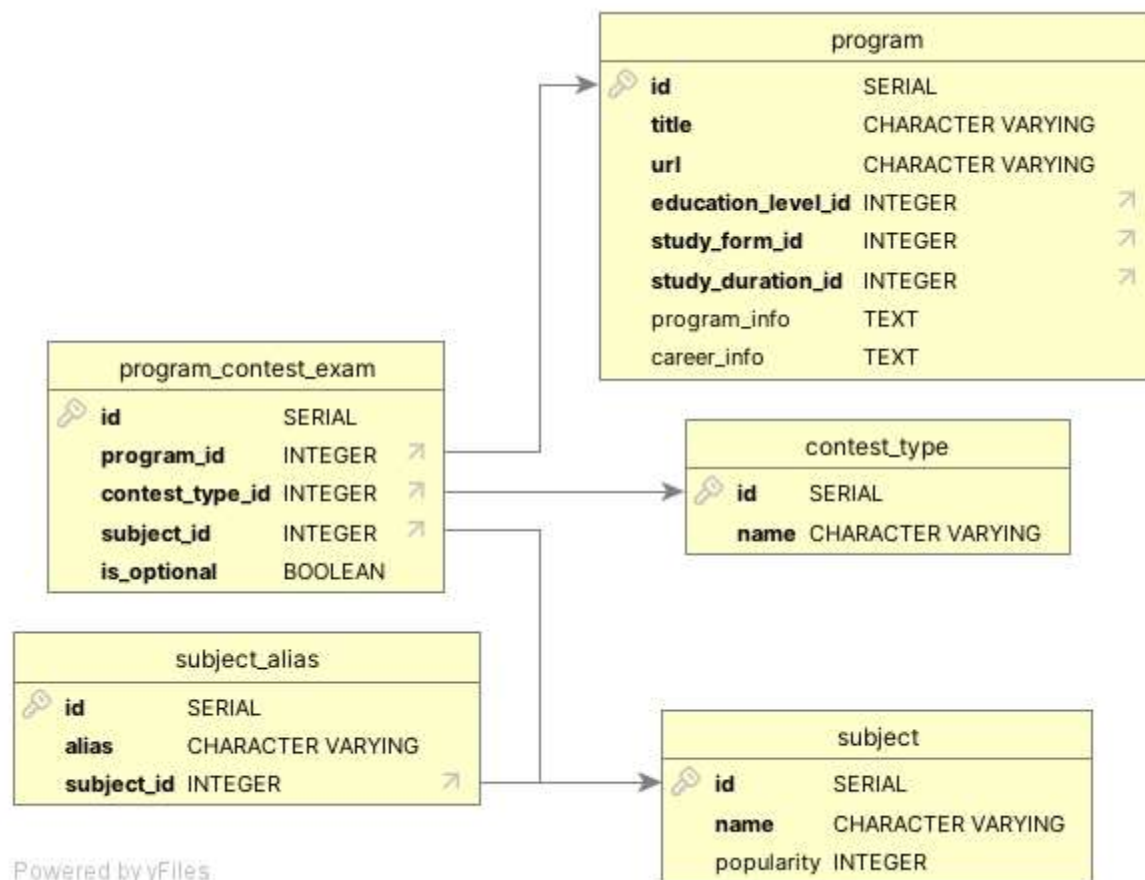


Powered by yFiles

База данных



База данных



Powered by yFiles

ЯЗЫКИ

Критерий	Python	C#	Java
Популярность (TIOBE, май 2025)	1 место (25,35%)	5 место (4,22%)	4 место (9,31%)
Популярность (GitHub Octoverse)	1 место	5 место	4 место
Асинхронность	Полная поддержка async/await	Полная поддержка async/await	Поддержка через CompletableFuture и аналоги
Библиотеки для Telegram-ботов	aiogram, python-telegram-bot, pyTelegramBotApi	Telegram.Bot, Telegram.BotAPI и др.	TelegramBots, java-telegram-bot-api и др.
Экосистема пакетов	Более 614,000 пакетов на PyPI	Более 100,000 пакетов на NuGet	Огромное количество, более 16,5 млн. пакетов на maven central repository
Машинное обучение	TensorFlow, PyTorch, scikit-learn, transformers	Transformers (портирован), ML.NET	Ограниченная поддержка, сторонние адаптации

Фреймворки

Критерий	python-telegram-bot	aiogram	pyrogram
Количество звёзд на GitHub	27,6 тыс.	5,1 тыс.	4,5 тыс. (архивирован)
Асинхронность	Да	Да	Да
Поддержка FSM	Да	Да	Нет
Поддержка webhook / polling	Да / Да	Да / Да	Да / Да
Middleware	Нет	Да	Нет
Интеграция с фреймворком компонент aiogram_dialog	Нет	Да	Нет
Обновления	Регулярные	Регулярные	Прекращены
Популярность / сообщество	Очень большое	Среднее	Было активное, сейчас – стагнация
Поддержка MTPROTO	Нет	Нет	Да
Подходит для обычных ботов	Да	Да	Да

СУБД

Критерий	PostgreSQL	SQLite	MySQL
Тип	Объектно-реляционная СУБД	Встроенная реляционная СУБД	Реляционная СУБД
Рейтинг DB-Engines	4 место	10 место	2 место
Звёзды на GitHub	17,6 тыс.	7,8 тыс.	11,4 тыс.
Популярность среди разработчиков	49% (StackOverflow Survey 2024)	Очень популярна в мобильной разработке	Широко используется
Производительность	Высокая	Высокая при чтении, особенно на малом объёме данных	Высокая, но как правило чуть меньше, чем у PostgreSQL
Параллелизм	Поддерживается	Ограниченный (блокировка на уровне файла)	Поддерживается

Векторные базы данных

Название	Лицензия	Фильтрация по метаданным	Особенности	Звезды GitHub (24.05.2025)
Qdrant [29]	Apache 2.0	Полноценная поддержка	gRPC/REST, Fast embeddings search, кластеризация	23,7 т. звезд
Pinecone [22]	Коммерческая	Частично (через API)	SaaS, легко масштабируется, без локального запуска	—
Weaviate [47]	Apache 2.0	Через GraphQL	модули для трансформерных моделей, встроенные модели	13,4 т. звезд
FAISS [11]	MIT	Нет встроенной поддержки	Библиотека от Meta, хороша для оффлайн-поиска, но не полноценная БД	35,1 т. звезд
Milvus [17]	Apache 2.0	Через JSON-фильтры	Поддержка кластеров,	34,9 т. звезд

Хранилища в оперативной памяти

Характеристика	Redis	Memcached	Dragonfly
GitHub звёзды	69,3 тыс.	13,8 тыс.	27 тыс.
Место в DB-Engines	1	4	35
Поддержка структур	Строки, списки, множества, хэши и др.	Только строки (ключ-значение)	Строки, списки, множества, хэши и др.
TTL (время жизни ключа)	Да	Д	Да
Pub/Sub	Да	Нет	Да
Персистентность	Да (RDB, AOF)	Нет	Да (улучшена)
Масштабируемость	Кластеры, репликация	Простая горизонтальная (шардирование вручную)	Многопоточность, shared-nothing
Совместимость	—	—	Совместим с Redis и Memcached
Зрелость проекта	С 2009 года	С 2003 года	С 2022 года
Поддержка Python	Официальная библиотека redis-py	Официальная библиотека rymemcache	Через совместимость с Redis

Модели

Модель (ссылка)	Параметры	Языки	Spearman (Encodechka STS)	Время инференса на CPU (среднее, с)
sergeyzh/LaBSE-ru-turbo [32]	128M	Русский	0,864	120,40
sergeyzh/LaBSE-ru-sts [31]	129M	Русский	0,845	42,84
sergeyzh/rubert-mini-sts [33]	32,4M	Русский	0,815	6,42
sergeyzh/rubert-tiny-sts [34]	29,4M	Русский	0,797	3,21
cointegrated/LaBSE-en-ru [4]	129M	Русский, Английский	0,794	42,87
Tochka-AI/ruRoPEBert-e5-base-512 [46]	139M	Русский	0,793	43,31
cointegrated/rubert-tiny2 [5]	29,4M	Русский	0,750	3,21