**Реферат**

**Слайд 1**

Добрый день, уважаемая комиссия. Меня зовут Ощепков Дмитрий Олегович. Представляю вашему вниманию дипломную работу на тему **"Разработка чат-бота в поддержку абитуриента ВятГУ"**, выполненную под руководством Разовой Елены Владимировны

**Слайд 2**

**Актуальность темы** исследования обусловлена необходимости повышения качества и доступности информационной поддержки абитуриентов ВятГУ

**Слайд 3**

**Цель работы** – разработка Telegram-бота для предоставления справочной информации и рекомендаций абитуриентам ВятГУ

**Слайд 4**

Для выполнения цели работы были выполнены следующие задачи

**Слайд 5**

При анализе предметной области у аналогов были выявлены следующие **преимущества**:

Возможность подписки на обновления

Поддержка нескольких языков

Доступ к официальным источникам

Свободный ввод и подключение оператора

Структурированная информация

Кроме того были у некоторых аналогов были выявлены следующие **недостатки**:

Отсутствие рекомендаций по программам

Переключение между клавиатурами

Избыточные визуальные элементы

Сильные и слабые стороны являлись ориентиром при написании работы.

**Слайд 6:**

Одной из ключевых задач при разработке чат-бота была **создание древовидной структуры** вопросов для удобной навигации и повышения релевантности ответов. Пример представлен на слайде

**Слайд 7:**

Итоговое приложение базируется на **докере** и имеет следующую структуру.

**Процесс запуска** выглядит следующим образом:

* Вначале разворачиваются независимые контейнеры: Data Base, Qdarant, Redis, Test Data Base и Test Redis
* 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 и Arq Worker (отложенные сообщения)

**Слайд 8:**

В рамках CI использован **github actions**, который создает тестовое окружение и запускает тестовый контейнер.

**Слайд 9:**

Для реализации основного компонента, бота, была выбрана **чистая архитектура**. Она разделяет приложение на 4 слоя:

* **Domain** содержит бизнес-объекты и независимую от внешних систем логику
* **Application** описывает бизнес-сценарии и использует интерфейсы вместо реализаций
* **Presentation** отвечает за взаимодействие с пользователем через Telegram и содержит интерфейсы к БД и сервисам
* **Infrastrucrure** включает конкретные реализации для работы с внешними системами

Чёткое разделение ответственности изолирует бизнес-логику от внешних технологий, упрощая замену компонентов, адаптацию и повторное использование ядра.

**Слайд 10:**

Для работы приложения была собрана **информация с официального сайта ВятГУ**, на основе которой была спроектирована реляционная база данных, представленная на рисунке.

Ключевые сущности здесь:

* program
* timeline\_event
* questions
* user

База данных содержит всю необходимую информацию для работы приложения и предоставляет к ней удобный доступ

**Слайд 11:**

На данном этапе с помощью бота **пользователь может**:

* Задать вопрос о поступлении
* Получить рекомендованные направления
* Настроить уведомления

**Слайд 12:**

На видео (22 с.) представлен сценарий **задания вопроса о поступлении** абитуриентом.

* Сейчас демонстрируется навигация с помощью категорий
* Теперь демонстрируется задание вопроса в свободной форме

**Слайд 13:**

Категории – это часто запрашиваемые данные, поэтому их лучше **закешировать**. На их примере продемонстрирвано взаимодействие с Redis

В начале бот смотрит в кеше, затем смотрит в базе данных

**Слайд 14:**

На этом слайде продемонстрировано **взаимодействие с микросервисом vector\_db\_service**.

Сервис выступает прослойкой между qdrant и основным приложением.

**Слайд 15:**

Рассмотрим сценарий **получения рекомендаций по направлениям** (видео 43 с.) :

(комментарии происходящего)

**Слайд 16:**

Последний реализованный сценарий – **настройка уведомлений** (видео 34 с.)

(комментарии происходящего)

**Слайд 17:**

В будущем **можно добавить**

* аналитику для администраторов
* автоматическое обновление данных
* веб-интерфейс на основе ядра приложения

**Слайд 18:**

В рамках данной работы **разработан телеграмм бот с гибкой архитектурой, базированной на докере**

Бот позволяет получать основную необходимую информацию для поступления в вуз

**Слайд 19:**

Спасибо за внимание

Время спокойного прочтения: 4.45

Видео: 1.39

Итого: 6.24