

инструмент воздействия на людей, следовательно, через архитектуру можно влиять на людей и, правильно применив особенности архитектуры, можно добиться невероятных высот в управлении социумом.

Библиографический список

1. Розов С.П. Петербуржец – особый тип человека/ С.П. Розов// URL: <https://el-tolstyh.livejournal.com/4292892.html> (дата обращения: 28.11.2019).
2. Храбрый И. С. Санкт-Петербург. Три века архитектуры/ И. С. Храбрый - С.-Петербург: ЗАО «Норинт», 2002. — 64 с.
3. Кауфман С.А. Архитектура Древнего Рима (Всеобщая история архитектуры)/ Кауфман С.А, Николаев И.С, Цирес А.Г, Блаватский В.Д. - Всеобщая история архитектуры, том. II, книга 2. М.: Издательство академии архитектуры СССР, 1948. - 437 с.
4. Розов С.П. Особенности энергетики Санкт – Петербурга и его жителей/ С.П. Розов// URL: <http://biomind.ru/bioenergy/petersburgers> (дата обращения: 28.11.2019).
5. Античность – архитектура древней Греции и Рима// URL: <https://meget.kiev.ua/arhitektura/antichnost/> (дата обращения: 28.11.2019).
6. Иванов А.Г. Римская архитектура IV-I вв. до н.э./ А.Г. Иванов// URL: https://studopedia.su/16_81138_rimskaya-arhitektura-IV-I-vv-do-n-e.html (дата обращения: 28.11.2019).
7. Архитектура древнего Рима// URL: https://ru.wikipedia.org/wiki/Архитектура_Древнего_Рима (дата обращения: 28.11.2019).
8. Архитектурный детерминизм// URL: <https://www.hisour.com/ru/architectural-determinism-28174/> (дата обращения: 28.11.2019).

УДК 20.15.05

Завгородний Н.И.

СОЗДАНИЕ ЧАТ БОТА ОРГАНИЗАТОРА CREATING AN ORGANIZER BOT CHAT

*Филиал «Протвино» государственного университета «Дубна»
Секция «Информационные технологии»*

Автор: Завгородний Никита Игоревич, студент 3-го курса направления «Информатика и вычислительная техника» филиала «Протвино» государственного университета «Дубна».

Научный руководитель: Кульман Татьяна Николаевна, кандидат технических наук, доцент кафедры информационных технологий филиала «Протвино» государственного университета «Дубна».

Author: Zavgorodniy Nikita Igorevich, 3rd year student of the direction "Informatics and computer engineering" of the branch "Protvino" state University "Dubna".

Scientific adviser: Kulman Tatiana Nikolaevna, candidate of technical sciences, associate professor of the department information technology of the branch "Protvino" state University "Dubna".

Аннотация

Рассматривается создание чат бота органайзера. Описываются выбор сервиса, проектирование базы данных, непосредственно разработка бота. В настоящее время бот эксплуатируется и доступен в социальной сети «ВКонтакте».

Abstract

The creation of a chat bot organizer is considered. Describes the choice of service, database design, bot development. Currently, the bot is operated and available on the «VKontakte» social network.

Ключевые слова: базы данных, органайзер, БД, чат бот, SQL, ВКонтакте, ВК, MySQL, MariaDB, PostgreSQL, кроссплатформенность, .NET CORE, Long Poll API.

Keywords: data base, organizer, DB, chat bot, SQL, VKontakte, VK, MySQL, MariaDB, PostgreSQL, cross-platform, .NET CORE, Long Poll API.

Чат бот – это результат взаимодействия человека и искусственного интеллекта: робот автоматически отвечает на сообщения пользователя. Чат боты используются для множества целей: для выполнения повседневных задач, развлечений и многих других. Размещают их в мессенджерах и социальных сетях, – Facebook Messenger, Telegram, «ВКонтакте», Slack и т.д.

Целью работы является создание чат бота органайзера для социальной сети «ВКонтакте».

Среднесуточная аудитория, по данным на август 2017 года, «ВКонтакте» составляет более 80 миллионов посетителей. По данным SimilarWeb в сентябре 2019 года, сайт «ВКонтакте» занимал 12 место по популярности в мире.

Актуальность: Тайм-менеджмент весьма сложная задача для многих современных и активных людей, умение управлять своим временем, зачастую, является секретом успешности. Стремительное развитие технологий и быстрота течения времени делают грамотное распределение времени серьёзной и актуальной проблемой.

Постановка задачи: создание чат бота органайзера с функционалом:

- создания обычных/ежедневных/еженедельных/ежемесячных/ежегодных событий через заданное время, в определённое время;
- удаления событий по названию, либо по идентификатору;
- поиска событий по времени/по названию;
- уведомления о созданных событиях.

Для хранения данных бот использует базу данных. Изначально разработка велась для СУБД MySql (Тестирование проходило на MariaDB), в последствии была так же добавлена поддержка PostgreSQL.

«ВКонтакте» для работы с ботами предоставляет на выбор Callback API и Vots Long Poll API. Гибкость разработки была в приоритете, поэтому выбор пал на Long Poll API.

Платформа .NET Core была выбрана из-за своей кроссплатформенности, - бота можно запустить на Windows, Linux и macOS. Благодаря такому выбору платформы можно создать полностью независимую версию бота, либо предварительно установить соответствующую версию .NET Core и использовать портативную (универсальную) версию консольного приложения.

Для взаимодействия с VK API используется библиотека VkNet, в ней реализованы методы VK API для .NET. Для взаимодействия с базой данных используются библиотеки MySql.Data и Npgsql.EntityFrameworkCore.PostgreSQL. Распознавание голосовых сообщений происходит благодаря API wit.ai.

Перед началом написания шаблона, были продуманы основные шаблоны сообщений, которыми пользователь будет оперировать при взаимодействии с ботом, они следующие:

1. Напомни мне через/в _time_ _event_
2. Через _time_ напомни мне _event_
3. Когда _event_
4. Через сколько _event_
5. Во сколько _event_
6. Удалить/удали _event_
7. Привет/начать
8. Помощь

Благодаря продумыванию сообщений, была спроектирована примерная логика анализа сообщений и форматы даты/времени, далее началась разработка.

При обработке используется поиск следующих форматов времени и даты:

Форматы даты:		Форматы времени:	
	1. 1 числа		1. 8 часов
	2. 01.01.2021		2. Восемь часов пять минут десять секунд
Во/В	3. 01.01	В	3. Час
	4. 01		4. Час дня 5 минут
	5. Вторник		5. 21^00^00
	6. Завтра		6. 21^00
			7. 8

Разработка началась с создания информативного и удобного для отладки консольного приложения, вся информация о функционировании бота выводится в консоль со временем и записывается в лог, для каждого сеанса создаётся свой лог файл, добавлены консольные команды для отладки и тестирования приложения.

Вид консольного приложения представлен на рисунке 1.

Рисунок 1. Вид консольного приложения

Был создан файл конфигурации в формате JSON, с этим форматом удобно работать в программе, и он легко читается пользователем, в него выведены обязательные переменные, вроде ключей доступа API и необязательные, вроде включения/отключения отладочных сообщений. Благодаря этому файлу, бота можно развернуть на любой системе за пару минут. При старте конфигурационный файл создаётся автоматически.

Файл конфигурации представлен на рисунке 2.

Далее была спроектирована база данных. Диаграмма базы данных MySQL представлена на рисунке 3. Для PostgreSQL она незначительно отличается типами данных и исполняемыми запросами, несмотря на это, при необходимости дампы базы данных из MySQL без особых усилий перенести в PostgreSQL.

Далее началась непосредственно разработка чата бота. Приложение использует один основной поток для ввода команд в консоль, для чата бота используются ещё два

потока: для принятия сообщений и для уведомления о событиях. Так же в зависимости от параметра в конфигурации, при получении сообщения, оно может отправиться для анализа в один из 16 потоков или анализироваться в этом же, последующие сообщения в обеих конфигурациях встанут в очередь, в многопоточном режиме работы очередь начнёт заполняться при загрузке всех 16 потоков, первое сообщение в очереди пойдёт в первый освободившийся поток.

Основной функцией приложения для анализа сообщения является Message.Analyze, логика анализа следующая:

1. Проверяем длину приложения.
 - a. Если длина некорректная, сообщаем об этом пользователю.

```
{
  "DebugMode": true,
  "ServerStatus": 1,
  "AccessToken": "",
  "GroupId": 0,
  "WetAiKey": "",
  "WaitEventsIntervalSec": 25,
  "SendEventsIntervalMs": 1000,
  "MultiThreadingForReceive": true,
  "DBEngine": "MySQL",
  "DBHost": "localhost",
  "DBPort": "3306",
  "DBLogin": "root",
  "DBPass": "",
  "DBName": "vkbot"
}
```

Рисунок 2. Файл конфигурации



Рисунок 3. Диаграмма базы данных

2. Если длина корректная, то удаляем знаки препинания с начала и конца предложения (при анализе они будут только мешать), разделяем сообщения на слова и проверяем первое слово на наличие слов вроде “Когда” или “Удали”.
 - a. При нахождении “Удали” удаляются шаблонные слова и словосочетания вроде “Удали”, “событие”, всё что осталось это id события или название события, передаём строку в следующий класс для удаления.
 - b. При нахождении “Когда” удаляются шаблонные слова и словосочетания вроде “Когда”, “у меня”, всё что осталось это название события, передаём строку в следующий класс для поиска.
3. Если слов не найдено, то удаляем лишние слова, вроде “Напомни мне”, “Создай событие”, ищем корректное слово “Через”, корректное – значит после него идёт время или “Сколько”.
 - a. При нахождении “Сколько” удаляем его, остальное – строка события, отправляем в следующий класс для поиска.
 - b. Если после “Через” стоит время, вырезаем правильно строку времени, остальное – строка события, отправляем обе строки в следующий класс для создания события.
4. Если “Через” не нашлось, ищем “Во вторник” или “Во сколько”.

- a. При нахождении “Во вторник” ищем время и правильно вырезаем строку времени, остальное – строка события, отправляем в следующий класс для создания события.
 - b. При нахождении “Во сколько” удаляем его, остальное – строка события, отправляем в следующий класс для поиска.
5. Далее ищем корректное “В”.
- a. При нахождении корректного “В” правильно вырезаем строку времени, остальное – строка события, отправляем в следующий класс для создания события.
6. Если ни одного корректного из слов “В”/”Во”/”Через” не нашлось, проверяем готовые шаблоны предложений, например: “Что ты умеешь”.
7. Если ничего не нашлось, сообщаем пользователю об ошибке.
- “Следующий класс” должен преобразовать и проверить строки на корректность и передать их в класс взаимодействия с БД и ждать ответа, отправить этот ответ пользователю.

Полную схему взаимодействия классов программы можно увидеть на рисунке 4.

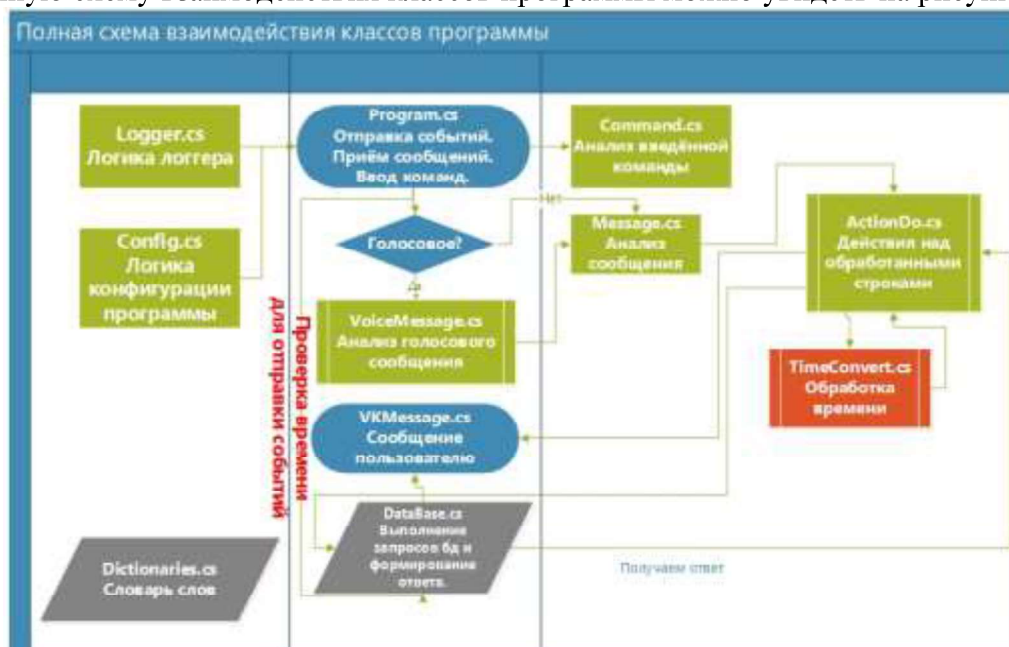


Рисунок 4. Схема взаимодействия классов программы

Бот был протестирован на Windows 10, Ubuntu 15.04, Ubuntu 18.04, были применены некоторые правки для корректной работы на разных ОС. Были привлечены несколько человек, каждый искал ошибки в работе, предлагал идеи. Все выявленные ошибки были исправлены. Некоторые предложения по функционалу были приняты и доработаны.

Результатом проделанной работы является:

- 1) В настоящее время бот создан, эксплуатируется, и доступен в группе по адресу <https://vk.com/vkorganizer>
- 2) Итоговое тестирование показало, что скорость ответа на текстовое сообщение составляет от 0,4с до 0,9с, в зависимости от конфигурации hardware и software (БД/ОС).
- 3) Бот поддерживает до 16 потоков для приёма сообщений.
- 4) Файл конфигурации позволяет легко изменять все параметры бота, такие как:
 - a) Режим отладки
 - b) Режим ответа/приёма сообщений

- c) Токен доступа
 - d) Id группы бота
 - e) Токен WetAi (Приём голосовых сообщений и их отключение)
 - f) Интервал проверки соединения с сервером
 - g) Интервал проверки сообщений для отправки
 - h) Многопоточность
 - i) Конфигурация БД
- 5) Спроектирована база данных, структура была записана в приложение. Поддержка СУБД с синтаксисом MySQL, PostgreSQL.
 - 6) Скорость запуска бота без ошибок до 2с, при ошибках, например, недоступности БД будет выполняться переподключение.
 - 7) Создание кроссплатформенного приложения портативного чат бота на платформе .NET Core с возможностью запуска на Windows, Linux и macOS.

Разработанный чат бот позволит использовать своё время эффективнее, грамотно составлять план своей деятельности, ставить цели и достигать своих целей в рамках предусмотренного времени. Чат бота можно использовать на любом устройстве/браузере, которые поддерживает социальная сеть “ВКонтакте”.

Библиографический список

1. Документация | Разработчикам -- VK API -- <https://vk.com/dev/manuals>
2. VKNET ВКонтакте API для .NET (C#) -- <https://vknet.github.io/vk>
3. NuGet Gallery | MySql.Data 8.0.18 -- <https://www.nuget.org/packages/MySql.Data/8.0.18/>
4. NuGet Gallery | Npgsql.EntityFrameworkCore.PostgreSQL 3.0.1 -- <https://www.nuget.org/packages/Npgsql.EntityFrameworkCore.PostgreSQL/3.0.1/>
5. Wit — HTTP API - <https://wit.ai/docs/http/20170307>

УДК 53.082.77

Капитонов И.Ю.

ПРИБОР ДЛЯ ДИАГНОСТИКИ МЕДИЦИНСКОГО ПУЧКА (ИОНИЗАЦИОННАЯ КАМЕРА) MEDICAL BEAM DIAGNOSIS DEVICE (IONIZATION CAMERA)

*Филиал «Протвино» государственного университета «Дубна»
Секция «Естественные и инженерные науки»*

Автор: Капитонов Илья Юрьевич, студент 4 курса направления «Физика» филиала «Протвино» государственного университета «Дубна».

Научный руководитель: Сытин Александр Николаевич, доктор физико-математических наук, заведующий кафедрой общеобразовательных дисциплин филиала «Протвино» государственного университета «Дубна».

Author: Kapitonov Ilya Yuryevich, 4d year student of the direction "Physics" of the branch "Protvino" state University "Dubna".

Scientific adviser: Sytin Alexander Nikolaevich, doctor of physical and mathematical sciences, head of department of general educational subjects of the branch "Protvino" state University "Dubna".

Аннотация