DOI: 10.12731/2227-930X-2021-11-3-57-67
УДК 004.93

РАЗРАБОТКА И ВНЕДРЕНИЕ СИСТЕМЫ МОНИТОРИНГА РАСПИСАНИЯ В ВЫСШИХ ОБРАЗОВАТЕЛЬНЫХ УЧРЕЖДЕНИЯХ ДЛЯ УДОБНОГО ВЗАИМОДЕЙСТВИЯ С ОБРАЗОВАТЕЛЬНЫМ ПРОЦЕССОМ

Биков Д.И., Хамидуллин М.Р.

Учебная отрасль – главная и самая широко распространенная отрасль. Развитие учебной системы в нашей стране не должно ограничиваться увеличением количества образовательных учреждений, оно также должно быть направлено на дальнейшее развитие с использованием информационных систем.

Для обеспечения комфортного мониторинга расписания в ВУЗах требуется ряд важных программных средств, упрощающих отслеживание учебной программы с использованием современных информационных систем.

***Цель** – создание системы мониторинга расписания в образовательных учреждениях для улучшения взаимодействия между студентами и образовательным процессом.*

***Метод или методология проведения работы:** в статье рассмотрен проект по отслеживанию расписания и взаимодействие с учебным процессом.*

***Результаты:** разработан публичный чат-бот в социальной сети ВКонтакте, в котором, по запросу, выдается, соответствующее требованию пользователя, ответ, связанный с образовательным процессом.*

***Область применения результатов:** полученные результаты целесообразно применять студентам, которые имеют возможность отслеживать список предстоящих учебных занятий и удобно взаимодействовать с ними.*

***Ключевые слова:** расписание; образовательный процесс; автоматизация; мобильная разработка; мониторинг*

DEVELOPMENT AND IMPLEMENTATION OF A SCHEDULE MONITORING SYSTEM IN HIGHER EDUCATIONAL INSTITUTIONS FOR CONVENIENT INTERACTION WITH THE EDUCATIONAL PROCESS

Bikov D.I., Khamidullin M.R.

The educational industry is the main and most widespread industry. The development of the educational system in our country should not be limited to an increase in the number of educational institutions, it should also be aimed at further development using information systems.

To ensure comfortable monitoring of the schedule in universities, a number of important software tools are required that simplify the tracking of the curriculum using modern information systems.

***Purpose** – creating a schedule monitoring system in educational institutions to improve the interaction between students and the educational process.*

***Method or methodology of work.** The article describes a project on schedule tracking and interaction with the educational process.*

***Results.** A public chatbot in the social network VKontakte has been developed, in which, upon request, a response related to the educational process is issued, corresponding to the user's request.*

***Scope of the results:** the results obtained should be applied to students who have the opportunity to track the list of upcoming training sessions and interact with them conveniently.*

***Keywords:** schedule; educational process; automation; mobile development; monitoring*

Введение

В данной статье рассматривается одно из возможных решений проблемы организации отслеживания и рассылки расписания. В настоящее время в России функционируют огромное количество высших образовательных учреждений, при этом важной особенностью является и остается удобность информационных систем.

Практически каждое учебное заведение, обладающая государственной аккредитацией стремится привлечь как можно больше студентов, поскольку страна нуждается в большом количестве высококвалифицированных специалистов. Важным условием для привлечения новых абитуриентов, помимо образования и дальнейшего трудоустройства, так же является грамотная реализация и ведение учебного процесса при помощи информационных систем

Разработка системы направлена на мониторинг расписания занятий в ВУЗах и удобное взаимодействие учебным процессом при помощи чат-бота социальной сети «ВКонтакте».

Материалы и методы

Чтобы создать бота, необходимо:

1. Сообщество, от имени которого чат-бот будет общаться с пользователями социальной сети «ВКонтакте».
2. Сервер, который будет получать уведомления о событиях.
3. Логика самого бота – скрипт, который определяет, как бот реагирует на то или иное событие [1].

Прежде всего, нужно продумать функциональность чат-бота. Во-первых, необходимо составить список возможных текстовых команд или событий, на которые бот должен реагировать, и соответствующие на них ответы бота. Чтобы реагировать на любые события, наш сценарий должен узнать о них [13]. В API «ВКонтакте» есть два подхода к этому – Callback API и Long Poll API.

При использовании CallBack API на стороне клиента создается сервер, который поддерживает непрерывную связь с серверами социальной сети «ВКонтакте». По созданному каналу сервер ожидает входных сигналов, а после получения, сразу приступает к обработке полученной информации. Кроме того, передача в данном типе осуществляется не пакетами, а по одиночке (рис 1). Таким образом создается значительная зависимость работоспособности бота от интернет-соединения [11].

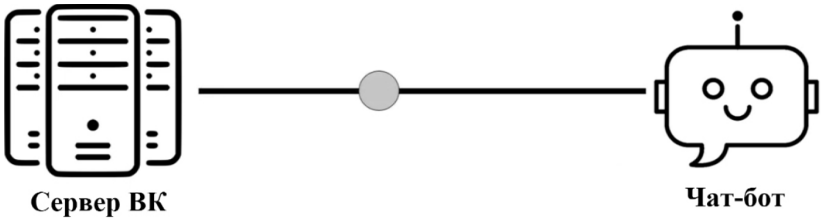

Рис. 1. Callback API. Непрерывная связь между клиентом и сервером

Второй способ получения обновлений – это подключение к Bots Long Poll API. При использовании Longpoll бот работает только на стороне клиента, ему не обязательно обращаться к серверам «ВКонтакте». Он сам при необходимости отправит сигнал на сервер ВК, после чего ожидает ответа, затем снова запрашивает. Данный тип API подразумевает передачу запросов от сервера не по одиночку, а пакетами, т.е. по несколько штук за одну передачу (рис 2). Это позволяет экономит трафик, потому что боту нет необходимости держать постоянную связь с сервером.

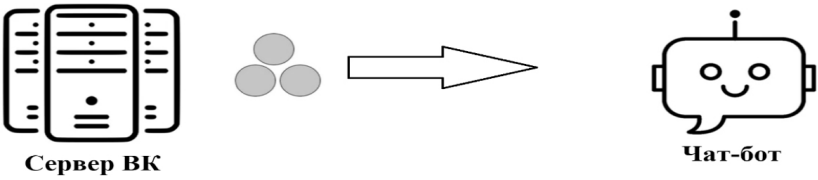

Рис. 2. Bots Long Poll API. Передача данных к серверу с помощью пакетов.

Можно сказать, что чат-бота можно создать на любом из API. Однако, Long Poll API больше подойдет для случаев, когда ботом одновременно пользуются небольшое количество людей [10]. Поскольку после получения сразу целого пакета новых событий, серверу необходимо все их обработать, это может вызвать сильную нагрузку на вычислительной мощности. Callback API напротив, позволяет распределить нагрузку на аппаратную часть, поскольку передача событий осуществляется сразу, однако требует постоянного стабильного подключения к интернету.

Реализация чат-бота

Первым шагом необходимо импортировать основную библиотеку vk_api [2]. Реализуется это при помощи команды import. Так же, нужно написать название библиотеки, которая используется в данном файле. Следующим шагом необходимо передать будущему коду ключ доступа к этой API, полученный из настроек сообщества ВКонтакте. Для того, чтобы сохранить в программе данный ключ, создается переменная token, в которой он будет храниться в виде текста. Когда программа знает ключ, можно переходить процессу авторизации. Для этого создается переменная с названием authorize, который и реализует подключение из библиотеки vk_api. В функции VkApi, в качестве аргументов передается ранее созданный ключ [3]. После всех манипуляций, необходимо выбрать тип API, который будет использоваться в дальнейшем и сообщить о своем выборе серверу. ВКонтакте предоставляет два типа API: «Bots Longpoll API» и «Callback API». Bots Longpoll API является более практичным, с точки зрения использования ресурсов хостинга. Дальнейшем шагом является создание переменной longpoll, и передача в аргумент функции переменную authorize. Поскольку бот должен работать постоянно, то все его функции должны находиться внутри бесконечного цикла. Использовать для этого необходимо цикл for в виде: «for event in longpoll.listen()». Функция listen, как можно понять из ее названия, «слушает» сервер ВКонтакте, то есть ожидает от него сообщение о каком-либо событии [4]. Далее необходимо проверить тип полученного сообщения. Так как добавлено условие, что тип сообщения должен быть текстом, бот будет видеть только текст из любого полученного сообщения. Далее можно переходить к смысловой обработке полученной информации. Начинается с того, что сохраним текст полученного сообщения. Для этого создается переменная под названием receive_message и пишется в качестве значения переменной функцию event.text [4]. Таким образом, в эту переменную будет автоматически записываться текст из каждого нового сообщения. Последующим действием является создание функции

write_message, который на вход будет получать 2 параметра: sender и message [9]. Для отправки сообщения пользователю реализуется метод messages.send, в котором необходимо передать серверу три обязательных параметра: id пользователя, текст и случайный идентификационный номер сообщения. Весь механизм логики ответов бота заключается в создании условных операторов. Выглядит это примерно так: «если текст полученного сообщения равен значению «Здравствуй», то тогда вызывается функция write_message, которому передается 2 параметра: id отправителя и текст ответа [5]. В заключении создается типовой ответ на случай, если то, что написал пользователь не совпадает с предусмотренными вариантами логики бота. Это реализуется при помощи конструкция else. Весь код программы представлен на рисунке 3.

```
import vk_api
from vk_api.longpoll import VkLongPoll, VkEventType
from vk_api.utils import get_random_id
def write_message(sender, message):
    authorize.method('messages.send', {'user_id': sender, 'message': message, 'random_id': get_random_id()})
token = "Здесь должен быть токен сообщества."
authorize = vk_api.VkApi(token = token)
longpoll = VkLongPoll(authorize)
for event in longpoll.listen():
    if event.type == VkEventType.MESSAGE_NEW and event.to_me and event.text:
        reseived_message = event.text
        sender = event.user_id
        if reseived_message == "Привет":
            write_message(sender, "Здравствуйте")
        elif reseived_message == "Пока":
            write_message(sender, "До свидания")
        else:
            write_message(sender, "Не понял")
```

Рис. 3. Код программы

Одной из ключевых функций бота является генерирование изображения с полным списком занятий, отсортированных по дням недели. Такое изображение удобно использовать для общего ориентирования. Эта функция требует некоторых пояснений.

Во-первых, для генерирования изображения боту необходимо прочитать и записать в память содержимое excel таблицы с расписанием. Каждый день недели расписание парсится с группы института в ВКонтакте, обрабатывается и сохраняется в отдельную переменную, что упрощает взаимодействие с данными в дальнейшем [14].

Далее производиться генерирования отдельных изображений для каждого из дней недели, в которых присутствуют занятия, посредством модуля Pillow. Так как этот модуль не поддерживает динамическое изменение параметров статичного изображения, высота изображения вычисляется заранее, исходя из высоты текста на основе заранее выбранного шрифта [6].

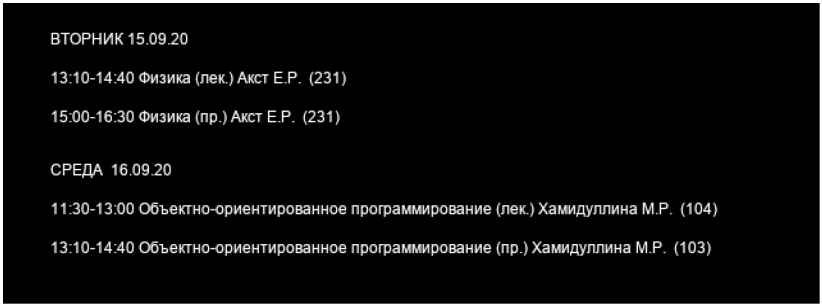

Рис. 4. Часть изображения с расписанием, созданное ботом

Далее все изображения собираются в одно, подобное представленному на рисунке 4, и итоговый результат отправляется в общую беседу группы во ВКонтакте либо индивидуально студенту либо преподователю.

Таким образом, создание общего изображения расписания посредством представленного фрагмента бота облегчает ориентирование в расписании.

Также снижается нагрузка на функциональную часть бота, так как используется сгенерированное, не связанное напрямую изображение, не требующее дополнительных вычислительных мощностей.

Если для работы локальной программы её достаточно скачать и установить, то бота для полноценной работы необходимо загрузить на удалённый сервер, называемый «хостингом» [15].

Хостинг представляет собой масштабированный удалённый компьютер, в основном на операционной системе Linux, предназначенный для работы сразу с большим количеством единиц программного обеспечения, требующего сетевое подключение [7].

Для работы был выбран регистратор доменных имён и хостинг REG.RU

После загрузки основных исполняемых файлов на хостинг следует настроить и запустить бота (рис. 5). Для этого используется SSH – сетевой протокол, использующийся для удалённого управления операционными системами.

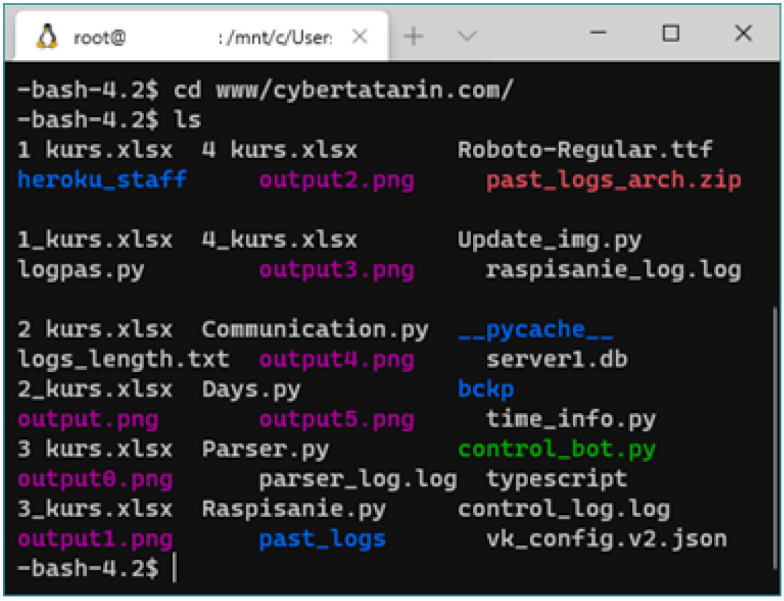

Рис. 5. Терминал сервера; подключение через SSH

Для работы чат-бота достаточно просто запустить исполняемые файлы посредством ввода команды «python Raspisanie.py & python Parser.py», но, если при этом закрыть SSH соединение, бот прекратит работу. Для решения этой проблемы была задействована программа Screen, по умолчанию встроенная в набор программ ОС Linux. Она позволяет запустить виртуальный терминал и уже в нём запустить процесс бота [8]. Такой терминал позволяет отключиться от сервера, сохраняя при этом работоспособность программного обеспечения.

Подводя итоги, можно сказать, что данная система активно используется в образовательном учреждении НЧФ КНИТУ-КАИ. Удобность системы мониторинга расписания протестировано и подтверждено достаточным количеством студентов института.

Список литературы

1. Акмаров П.Б. Кодирование и защита информации: учебное пособие. Ижевск: Ижевская ГСХА, 2016. 136 с. https://e.lanbook.com/book/133975 (дата обращения: 20.12.2020).
2. Богачёв К.Ю. Основы параллельного программирования. Москва: БИНОМ. Лаборатория знаний, 2015. 345 с. https://ibooks.ru/reading.php?productid=350082 (дата обращения: 20.12.2020).
3. Подбельский В.В. Язык декларативного программирования XAML. Москва: ДМК Пресс, 2018. 336 с. https://e.lanbook.com/book/111428 (дата обращения: 20.12.2020).
4. Рудинский И.Д. Технология проектирования автоматизированных систем обработки информации и управления. Москва: Горячая линия-Телеком, 2011. 304 с. https://e.lanbook.com/book/5191 (дата обращения: 20.12.2020).
5. Соколова В.В. Разработка мобильных приложений: учебное пособие. Томск: ТПУ, 2014. 176 с. https://e.lanbook.com/book/82830 (дата обращения: 20.12.2020).
6. Тугов В.В. Проектирование автоматизированных систем управления : учебное пособие / В.В. Тугов, А.И. Сергеев, Н.С. Шаров. Санкт-Петербург: Лань, 2019. 172 с. https://e.lanbook.com/book/123695 (дата обращения: 20.12.2020).
7. Шаньгин В.Ф. Защита информации в компьютерных системах и сетях: учеб. пособие. Москва: ДМК Пресс, 2012. 592 с. https://e.lanbook.com/book/3032
8. Khamidullin M.R., Mardanshin R.G., Prozorov A.V., Karimov R.I. The Introduction of QR -Codes in Production Processes // Journal of Environmental Treatment Techniques. Special Issue on Environment, Management and Economy, 2019. P. 1097-1100.

9. Isavnin A.G., Khamidullin M.R. Determining of total expenses for the objective of equipment replacement // Life Science Journal. 2014, №11 (6). P. 704-706.

References

1. Akmarov P.B. *Kodirovanie i zashchita informatsii: uchebnoe posobie* [Coding and protection of information]. Izhevsk: Izhevskaya GSKhA, 2016, 136 p. https://e.lanbook.com/book/133975
2. Bogachev K.Yu. *Osnovy parallel'nogo programmirovaniya* [Fundamentals of Parallel Programming]. Moscow: BINOM. Laboratoriya znaniy, 2015, 345 p. https://ibooks.ru/reading.php?productid=350082
3. Podbel'skiy V.V. *Yazyk deklarativnogo programmirovaniya XAML* [XAML declarative programming language]. Moscow: DMK Press, 2018, 336 p. https://e.lanbook.com/book/111428
4. Rudinskiy I.D. *Tekhnologiya proektirovaniya avtomatizirovannykh sistem obrabotki informatsii i upravleniya* [Design technology for automated information processing and control systems]. Moscow: Goryachaya liniya-Telekom, 2011, 304 p. https://e.lanbook.com/book/5191
5. Sokolova V.V. *Razrabotka mobil'nykh prilozheniy: uchebnoe posobie* [Mobile Application Development]. Tomsk: TPU, 2014, 176 p. https://e.lanbook.com/book/82830
6. Tugov V.V., Sergeev A.I., Sharov N.S. *Proektirovanie avtomatizirovannykh sistem upravleniya : uchebnoe posobie* [Design of automated control systems]. St. Petersburg: Lan, 2019, 172 p. https://e.lanbook.com/book/123695
7. Shan'gin V.F. *Zashchita informatsii v komp'yuternykh sistemakh i setyakh: ucheb. posobie* [Information protection in computer systems and networks]. Moscow: DMK Press, 2012, 592 p. https://e.lanbook.com/book/3032
8. Khamidullin M.R., Mardanshin R.G., Prozorov A.V., Karimov R.I. The Introduction of QR -Codes in Production Processes. *Journal of Environmental Treatment Techniques. Special Issue on Environment, Management and Economy*, 2019, pp. 1097-1100.
9. Isavnin A.G., Khamidullin M.R. Determining of total expenses for the objective of equipment replacement. *Life Science Journal*, 2014, no. 11 (6), pp. 704-706.

ДАННЫЕ ОБ АВТОРАХ

Биков Данир Инсафович, студент

Набережночелнинский филиал Казанского национального исследовательского технического университета им. А.Н. Туполева

Академика Королева, 1, г. Набережные Челны, 423814, Российская Федерация

danir.bikov@gmail.com

Хамидуллин Марат Раисович, доцент, кандидат экономических наук

Набережночелнинский филиал Казанского национального исследовательского технического университета им. А.Н. Туполева

Академика Королева, 1, г. Набережные Челны, 423814, Российская Федерация

nayka_prom@mail.ru

DATA ABOUT THE AUTHORS

Danir I. Bikov, student

Kazan National Research Technical University, Branch in Naberezhnye Chelny

1, Akademika Koroleva Str., Naberezhnye Chelny, 423814, Russian Federation

danir.bikov@gmail.com

Marat R. Khamidullin, PhD in Economics

Kazan National Research Technical University, Branch in Naberezhnye Chelny

1, Akademika Koroleva Str., Naberezhnye Chelny, 423814, Russian Federation

nayka_prom@mail.ru

ORCID: 0000-0002-3326-0955