Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра «Информатика»

УТВЕРЖДАЮ

Заведующий кафедрой

_____А.С. Кузнецов

« » июля 2019 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.04 - Программная инженерия

Разработка программного бота формирования расписания образовательного процесса в учебном заведении

Руководитель	_____	старший преподаватель	А. С. Михалёв
Выпускник	_____		А. М. Отто
Консультант	_____	доцент, канд.техн.наук	А. С. Кузнецов
Нормконтролер	_____	доцент, канд.техн.наук	О. А. Антамошкин

Красноярск 2019

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра «Информатика»

УТВЕРЖДАЮ
Заведующий кафедрой
_____А.С. Кузнецов
« » июля 2019 г.

ЗАДАНИЕ

НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

в форме бакалаврской работы

Студент Отто Андрей Михайлович

фамилия, имя, отчество

Группа КИ15-17Б Направление (специальность) 09.03.04

номер код

Программная инженерия

наименование

Тема выпускной квалификационной работы: Разработка программного бота формирования расписания образовательного процесса в учебном заведении.

Перечень разделов ВКР: введение, описание предметной области, использованные технологии и платформы при разработке, архитектура проекта, демонстрация работы.

Перечень графического материала: Презентационные слайды PowerPoint.

Руководитель ВКР А. С. Михалев, старший преподаватель, ИКИТ СФУ

инициалы, фамилия, должность, ученое звание и место работы

подпись инициалы и фамилия

Задание принял к исполнению

подпись, фамилия и инициалы студента

РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка программного бота формирования расписания образовательного процесса в учебном заведении» содержит 38 страниц текстового документа, 34 иллюстрации, 1 таблица, 16 использованных источников.

КЛЮЧЕВЫЕ СЛОВА: РАСПИСАНИЕ, PYTHON, TELEGRAM, БОТ, ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА БОТА ДЛЯ МЕССЕНДЖЕРА.

Цель выпускной квалификационной работы: проектирование и разработка бота для мессенджера Telegram, который по запросу формирует расписание на студентов СФУ.

Для достижения поставленной цели необходимо реализовать следующие задачи:

− исследовать наиболее значимую литературу в выбранной области разработки;

− проанализировать возможные средства получения студентами информации о расписании;

− проанализировать предметную область;

− спроектировать и разработать бота для мессенджера.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Для успешной учебной детальности студенту нужно грамотно планировать свое время. Поэтому ему необходимы удобные инструменты для отслеживания своих занятий. Студенты уже имеют возможность найти расписание вручную на главном сайте СФУ или же воспользоваться приложением «Студент СФУ». В данной дипломной работе был разработан и описан новый способ – чат-бот для мессенджера Telegram, который предоставляет студентам СФУ актуальное для них расписание занятий.

Для реализации бота необходимо определить несколько моментов:

– язык программирования;

– сервис для хостинга бота;

– сбор данных;

– хранение данных;

– сервис контроля версий;

– сервис для отслеживания задач.

Также следует определить какие возможности будет реализовывать бот:

– поиск расписания по введённому пользователем сообщению;

– выдача расписания по командам от пользователя, которые можно настроить для бота через мессенджер;

– поиск расписания преподавателей;

– подписка на ежедневное уведомление о завтрашнем расписании.

1 Описание предметной области

Бот – специальная программа, которая берёт на себя роль человека. Например, боты могут быть автоматическими собеседниками в чатах в службах поддержки или в нужный момент размещать от вашего имени ставки на интернет-аукционах. Боты могут давать информацию о погоде, афишу мероприятий, подобрать ресторан. Также боты используются в технической поддержке пользователей по простым вопросам [1].

Боты довольно популярное дополнение у компаний при работе с клиентами. Так по прогнозу Business Insider, уже к 2020 году 80 % компаний будет пользоваться чат-ботами [2]. Ничто не мешает учебным заведениям в рамках образования и ведения учебного процесса также внедрять ботов для студентов и преподавателей.

Все популярные мессенджеры (Telegram, Viber, WhatsApp) и популярные социальные сети (ВКонтакте, Facebook) поддерживают создание ботов для своих платформ.

По данным исследования аудитории от Telegram Analytics на 2019 год ботами в мессенджере Telegram пользуется больше 40% пользователей (рисунок 1.1) [3].

Чат-бот для предоставления расписания занятий для студентов отлично подходит под изначальную идею ботов, как о автоматизированных помощниках внутри компаний, в частном случае в университете.

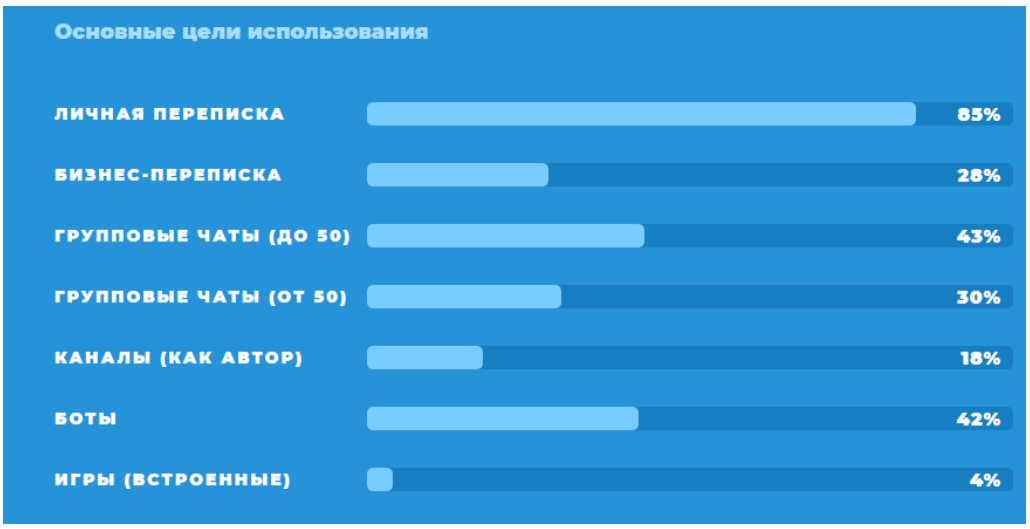

Рисунок 1.1 – Для чего пользователи используют Telegram

1.1 Аналоги

В данный момент для студента СФУ существует несколько способов узнать свое расписание. Рассмотрим их подробнее.

1.1.1 Найти расписание на сайте своего института. У каждого института в СФУ есть свой сайт, а на нем обязательно можно найти раздел посвящённый расписанию групп этого института. Обычно это просто файл расширения .xlsx, который нужно скачать (рисунок 1.2), так и он еще внутри содержит таблицу с несколькими группами, где нужно найти свою. Тоже не очень удобно и наглядно каждый раз заглядывать в файл или заходить на сайт института.

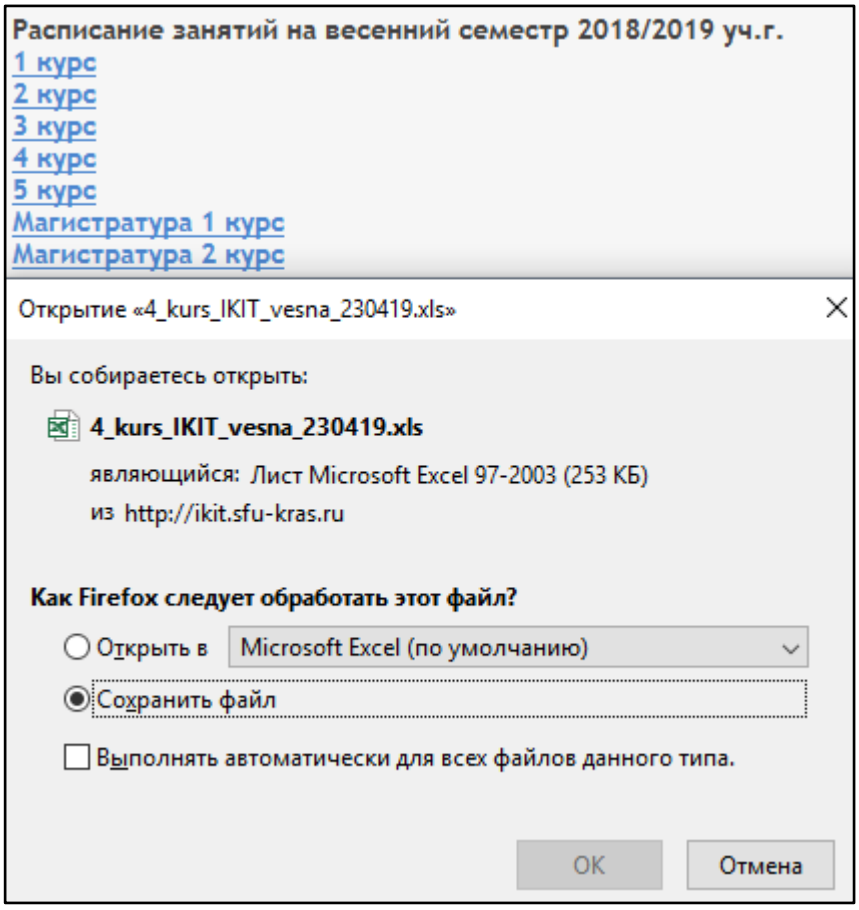

Рисунок 1.2 – Процесс просмотра расписания на сайте ИКИТ

1.1.2 Раздел с расписанием на главном сайте СФУ. Есть поиск по группам, также группы разбиты на институты, что проще искать. Страницу с расписанием своей группы можно добавить в избранное и каждый раз обращаться к ней из браузера. Также есть подсветка текущей недели. Проблемой является отсутствие адаптивности сайта под расширение смартфонов, что не очень удобно, когда просматриваешь расписание: таблица получается большой и нужно проделать много движений пальцем, чтобы добраться до нужного дня (рисунок 1.3).

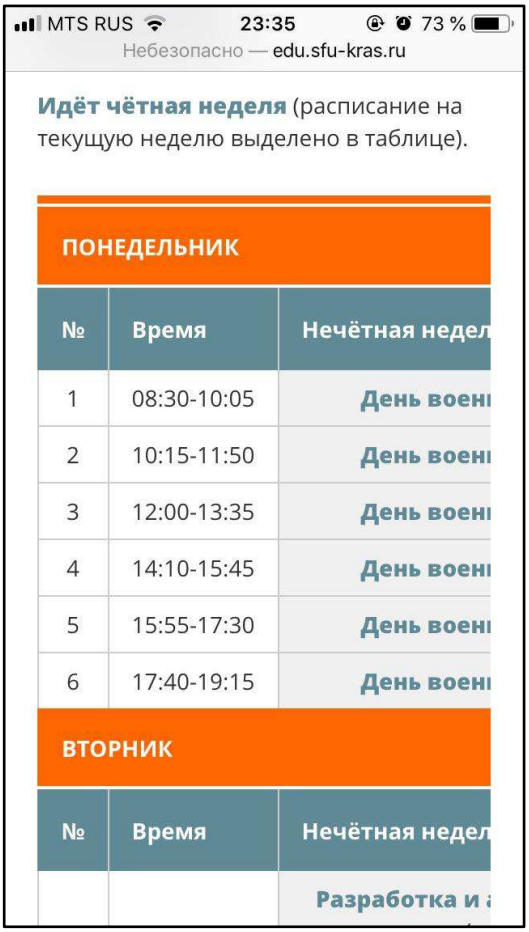

Рисунок 1.3 – Расписание на сайте СФУ при вертикальном расположении смартфона

1.1.3 Приложение «Студент СФУ». На главной странице приложения отображается расписание группы на сегодня: время, название предмета, аудитория, преподаватель. Также можно движением пальца менять главный экран для просмотра расписания на завтра и последующие дни. Минусом является, то что для каждой платформы (Android, iOS) приходится разрабатывать отдельно приложение, выпускать отдельные обновления, а также приложение занимает память на смартфоне.

1.2 Сравнение аналогов

Использование бота для расписания имеет ряд преимуществ по сравнению с аналогами, описанными ранее. Так боту не нужна отдельная установка, в

отличии от приложения, что экономит пользователю память на устройстве. Пользователю не требуется отдельно обновлять бота в отличии от приложения. Бота не требуется переносить на несколько платформ, так как мессенджер или социальная сеть уже поддерживает все операционные системы.

Для наглядно сравнения была составлена таблица 1.

Таблица 1 – Сравнение аналогов

Характеристики оценивания	Расписание на сайте института	Расписание на сайте СФУ	Приложение «Студент СФУ»	Бот
Отдельная установка	Не требуется	Не требуется	Требуется	Не требуется
Скачивание обновлений	Не требуется	Не требуется	Требуется	Не требуется
Одна версия на все ОС	Да	Да	Нет	Да
Удобное отображение	Нет	Нет	Да	Да
Наличие мобильной версии	Да (браузер)	Да (браузер)	Да	Да
Наличие ПК версии	Да (браузер)	Да (браузер)	Нет	Да (настольная версия Telegram)
Возможность расширения другим функционалом	Нет	Да	Да	Да

1.3 Актуальность

Учитывая предыдущие подразделы можно сказать что разработка бота является актуальной. Студентам необходим альтернативный источник получения расписания. Бот имеет свои плюсы по сравнению с существующим приложением и просто сайтом с расписанием. Также в будущем бот может приобрести и другие функции, не связанные с расписанием, например, рассылка новостей, оповещение о грядущих мероприятиях. Также бота можно подключать к различным чатам в границах мессенджера.

2 Использованные технологии и платформы при разработке

Кратко обо всём:

– Язык программирования: Python;

– Обработка запросов: фреймворк Flask;

– Парсинг данных: библиотека BeautifulSoup4;

– Сервис контроля версий: GitHub (OttoAndrey/TimetableSFUBot);

– Сервер размещения бота: Heroku;

– Хранение данных: СУБД PostgreSQL;

– ПО для управления проектом: Trello.

Далее в главе каждый из пунктов будет описан более подробно.

2.1 Язык программирования

Бота можно разработать на разных языках: JavaScript, C#, Java и других. Но для разработки данного бота был выбран Python. У данного языка простой синтаксис, много вспомогательных библиотек для парсинга страниц и обработки запросов, например, Flask или Django, большая аудитория среди программистов, что позволяет проще найти ответы на вопросы на различных форумах. Так в рейтинге IEEE Spectrum за 2018 год Python является первым по популярности языком [4]. А в рейтинге TIOBE Index за май 2019 года язык Python входит в первую пятерку популярных языков программирования [5].

2.2 Система контроля версий

Проект был опубликован на сервисе по контролю версий GitHub. Сервис позволяет публиковать проект как в открытых, так и в закрытых репозиториях. Преимуществом сервиса является то, что он запоминает каждую версию проекта и если происходит критический баг в новой версии, то всегда можно откатить

проект к более старому-стабильному варианту. Проект можно найти по адресу OttoAndrey/TimetableSFUBot.

2.3 Обработка запросов к боту

Каждое сообщение, которое приходит боту от пользователей, это post-запрос. Для того чтобы бот мог их принимать и обрабатывать был использован микрофреймворк Flask. Обычно Flask используют для создания простых веб-приложений, но в данном случае его удобно использовать для получения запросов от пользователей. Так декоратор «route» [6] над функцией index() позволяет получать post-запросы (рисунок 2.1).

```
705  @app.route('/', methods=['POST', 'GET'])
706  def index():
707      if request.method == 'POST':
708          r = request.get_json()
709          chat_id = r['message']['chat']['id']
710          message = r['message']['text']
711
712          user_massages_handler(chat_id, message)
713
714          return jsonify(r)
715      return 'kek'
```

Рисунок 2.1 – Декоратор @app.route

2.4 Сбор данных

Для сбора данных сайта СФУ были использованы библиотеки requests и BeatifulSoup4. Библиотека requests позволяет получить html-текст страницы по указанному url-адресу. Чтобы обработать полученные данные и взять только часть из них, например, класс table или заголовок h3, в котором располагается расписание, используется библиотека BeautifulSoup4.

2.5 Сервер для размещения

Чат-бот размещен на сервисе Heroku. Данный сервис позволяет синхронизировать проект с аккаунтом на GitHub, что позволяет загружать и обновлять проект в один клик мыши. Также можно включить авто обновление проекта, когда проект будет обновлен на GitHub он автоматически загрузится на Heroku.

Heroku позволяет загружать приложения по бесплатной подписке, но имеет свои ограничения по сравнению с платными версиями. Например, сайт или бот засыпает после 30 минут, если не получает запросов. Поэтому первый запрос пользователя во время сна бота приходит с задержкой в несколько секунд. Данное ограничение можно обойти, использовав сервисы UptimeRobot [7] или Kaffeine. В данном проекте был использован Kaffeine – это одностраничный сайт, который принимает ссылку на проект и отправляет к нему запросы каждые 30 минут, не давая боту уснуть (рисунок 2.2).

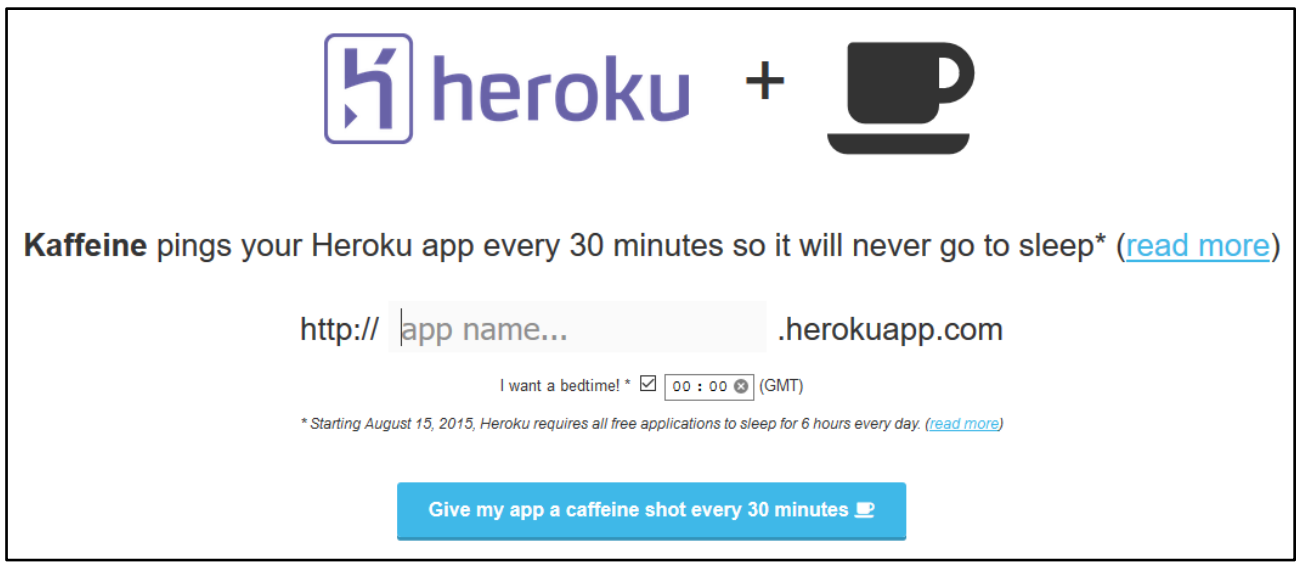

Рисунок 2.2 – Kaffeine очень прост в использовании

Еще одно ограничение бесплатной версии Heroku, это ограниченное количество часов работы приложений. Суммарно на весь аккаунт даётся 550

часов. Если привязать банковскую карту, то добавляется еще 450 часов в месяц. Суммарно 950 часов с запасом хватает на месяц для бесперебойной работы бота.

Также в Heroku реализовано множество вспомогательных расширений для хранения данных, мониторинга приложения или тестирования приложения. Так в данном проекте используется Heroku Postgres [8] для хранения данных. Данный сервис создаёт базу данных и таблицы, а также позволяет формировать sql-запросы, называемые DataClips, которые можно сохранять, чтобы пользоваться ими в любой удобный момент (рисунок 2.3).

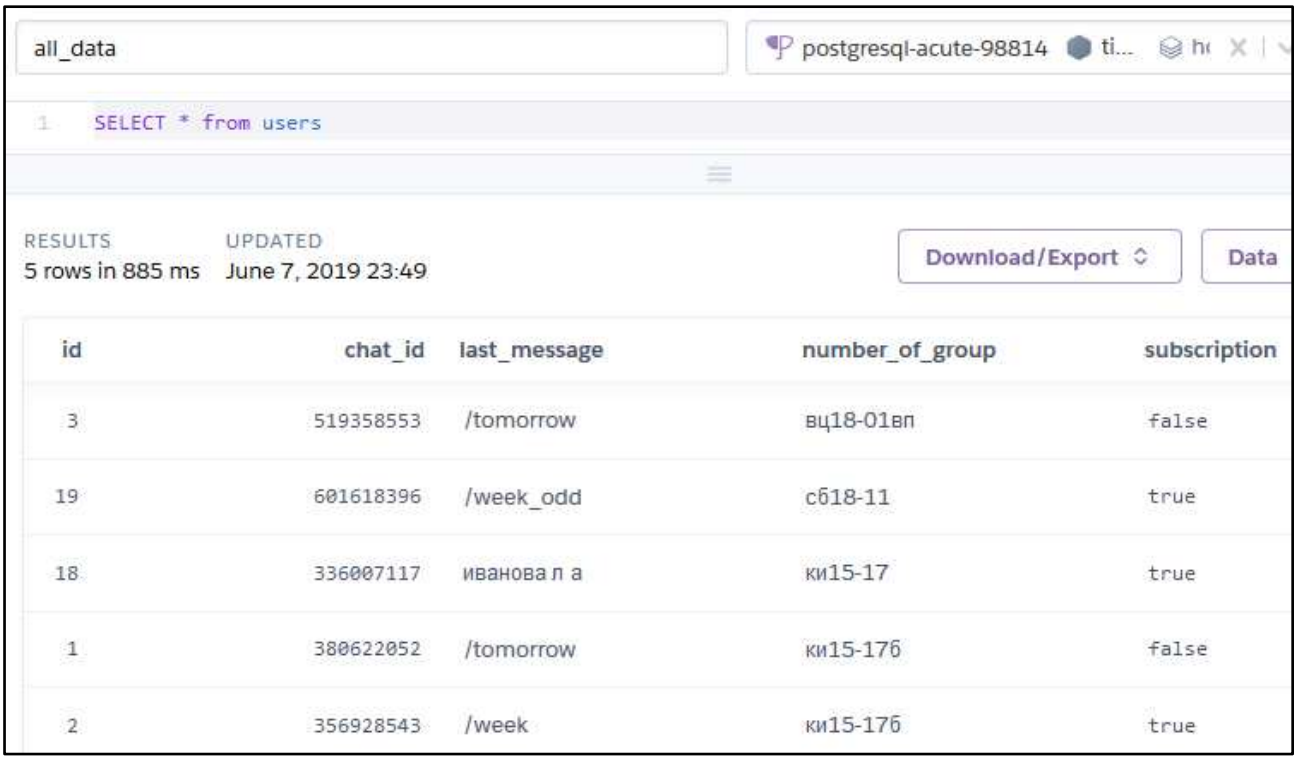

Рисунок 2.3 – Простой запрос по всем зарегистрированным пользователям

2.6 Мессенджер для бота

Telegram – один из популярных мессенджеров в мире разработанный нашим соотечественником Павлом Дуровым.

Выбор именно этого мессенджера обусловлен тем, что он является самым безопасным: не имеет утечек данных в отличии от WhatsApp [9] и не сливает

данные о пользователях силовым структурам как тот же WhatsApp [10] или ВКонтакте [11].

Хоть Telegram и проигрывает другим мессенджерам по количеству пользователей в России [12] всё же безопасность данных пользователей, в данном случае студентов и преподавателей, является приоритетнее. Если уж и привлекать пользователи пользоваться теми или иными платформами или продуктами, то только наиболее безопасными.

К сожалению, Telegram заблокирован на территории России, но суммарно 95% пользователей не испытывают проблем в использовании Telegram (рисунок 2.4) [3]. Так почти половина не использует средств обхода блокировки и у них всё работает. Остальные 45% так или иначе используют прокси или vpn для доступа к Telegram. И только у 5,4% пользователей Telegram, которые не используют средства обхода приложение периодически работает с перебоями.

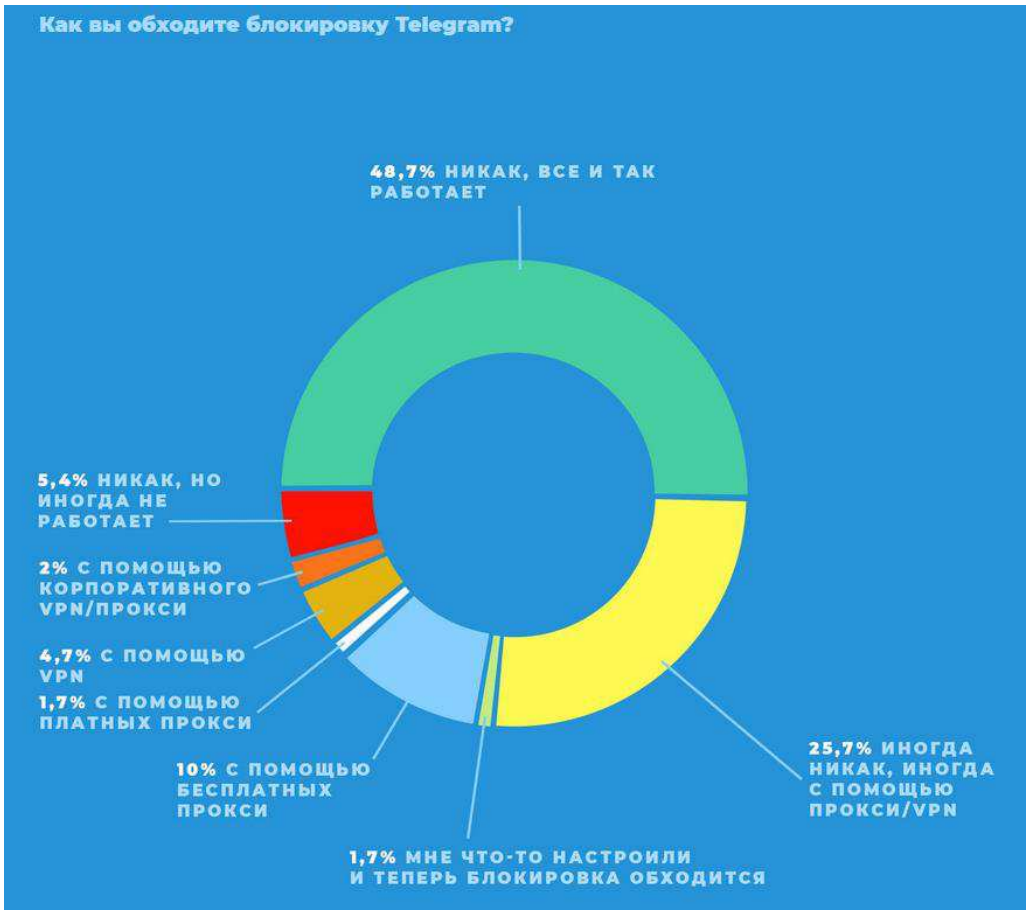

Рисунок 2.4 – Работоспособность Telegram будучи заблокированным в России

2.7 Организация разработки бота с помощью Trello

Для более удобной связи между членами команды в работе над проектом можно использовать программное обеспечение для управления проектами или систему задач. Такая программа поможет в управлении над проектом в общем, связав руководителей, дизайнеров, программистов и тестеров, помогающая им взаимодействовать между собой и отслеживать вклад и действия каждого. Также удобно при проекте создать узконаправленную ветвь для любой группы.

Хоть бот и разрабатывался одним человеком всё равно удобно использовать Trello для записи идей и отслеживания их выполнения. Данная система является бесплатной, имеет приятный интерфейс и интуитивно понятна. Trello позволяет максимально удобно и гибко распределить задачи в команде над проектом. Интерфейс программы сделан в виде досок со списками, на которых располагаются карточки (рисунок 2.5) Каждая карточка - это отдельное задание для выполнения. Каждое задание начинает со списка «Сделать» после того, как участник команды приступает к карточке она переходит на список «В процессе». После выполнения карточка переходит на список «Готово». Также можно создать дополнительные, например, «Возможно придется обновить». Таким образом очень удобно визуально понимать необходимый перечень работ, который надо выполнить, которые в процессе выполнения и выполненные задачи.

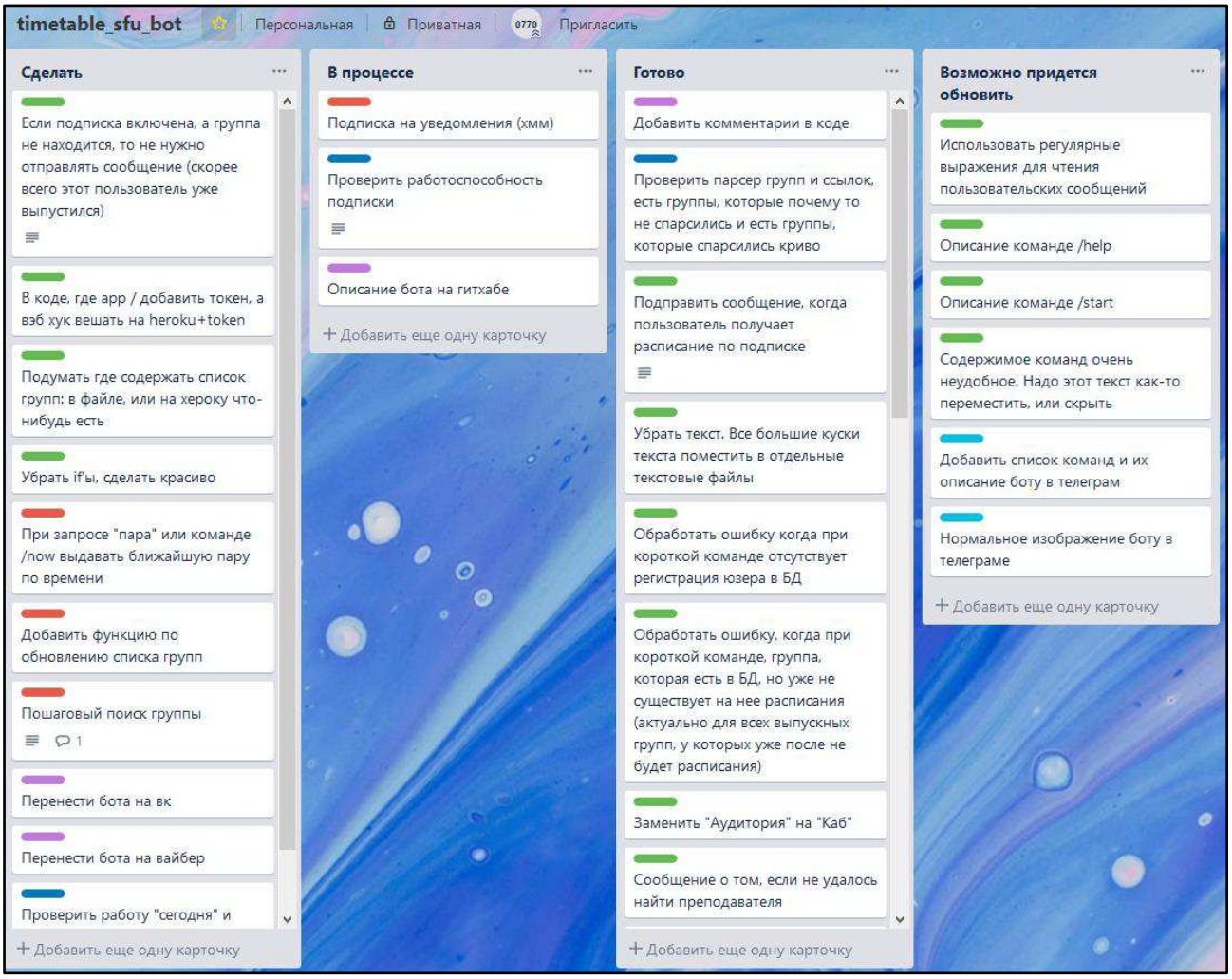

Рисунок 2.5 – Доска при разработке чат-бота

Также каждая карточка имеет цветную метку (рисунок 2.6), которая помогает определить к какому типу задачи относится та или иная карточка. Trello позволяет добавлять свои метки и настраивать им цвет.

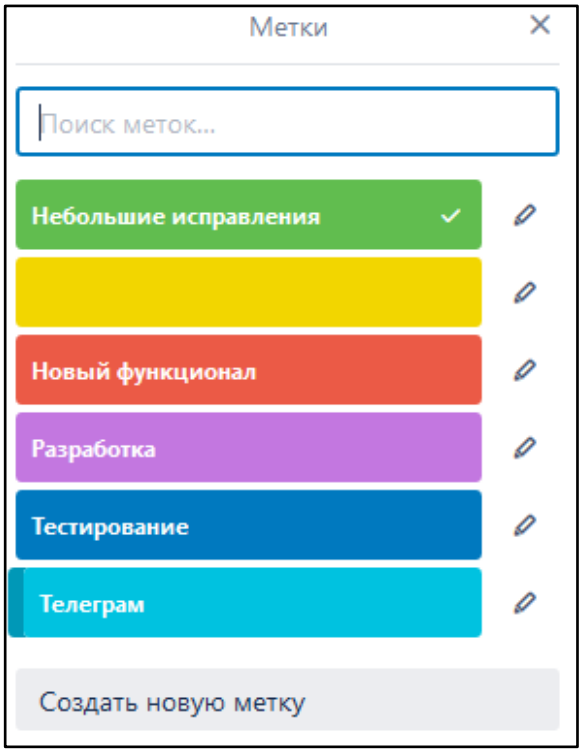

Рисунок 2.6 – Настраиваемые метки

3 Архитектура проекта

В данной главе описана архитектура проекта и его функции. Также описано хранение данных: таблицы, типы данных. Описаны алгоритмы различных возможностей бота.

3.1 Шаблон проектирования

Для данного проекта был использован популярный шаблон MVC. Благодаря этому шаблону проект состоит из трех основных частей (рисунок 3.1), каждая из которых содержит функционал, отвечающий за реализацию определённой логики [13].

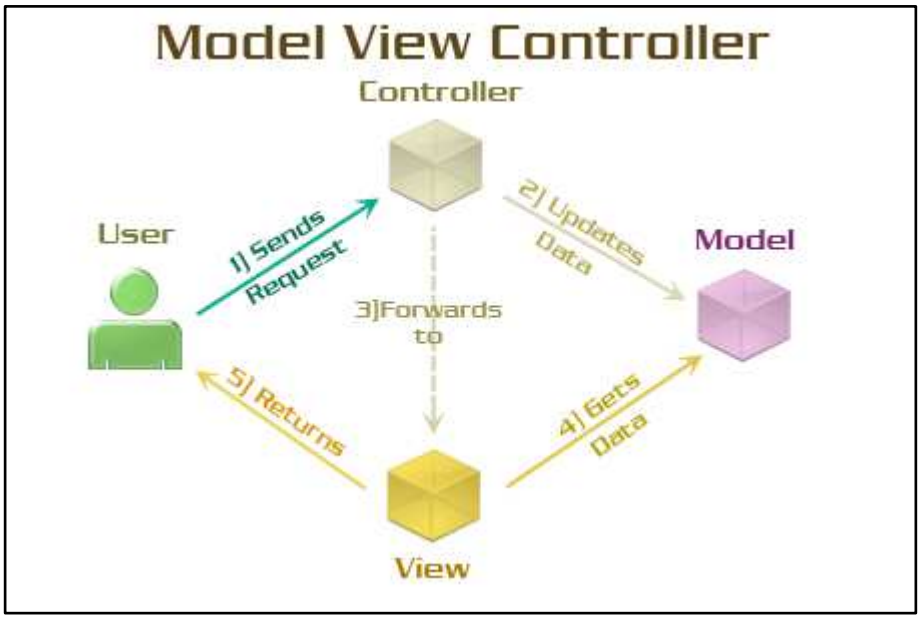

Рисунок 3.1 – Шаблон MVC

3.1.1 Модуль Controller получает сообщение пользователя и определяет ответ для него. Также содержит функции, которые не возвращают конечный результат для пользователя, а служат как вспомогательные при построении ответа. Его функции:

– user_massages_handler(chat_id, message) – главная функция модуля. На воход получает чат-айди пользователя и его сообщение. Благодаря регулярным выражениям определяет введённую пользователем команду и решает какую функцию из модуля View использовать для ответа;

– get_html(url) – функция для получения html-текста страницы. На вход получает адрес. Возвращает html-текст страницы;

– get_group_url(number_of_group) – функция для получения адреса страницы с расписанием. На вход получает номегр группы и по этому номеру производит поиск в базе данных. Возвращает url страницы;

– get_teacher_url(message) – аналогично функции get_group_url() только для формирования адреса для преподавателя;

– get_text_of_command(message) – такие команды как /start и /help содержат достаточно большое количество текста, поэтому было решено поместить их в отдельный текстовый файл. Данная функция на вход получает соответствующую команду, а возвращает ее текст;

– send_every_day_timetable() – функция ответственная за рассылку ежедневного расписания;

– schedule_run() – функция, которая содержит цикл для запуска объекта schedule, который в свою очередь запускает функцию send_every_day_timetable();

– send_message(chat_id, text='Не удалось найти группу') – функция, которая отправляет сообщение пользователю с сформированным расписанием или сообщение об ошибке.

3.1.2 Модуль View содержит функции, которые возвращают конечный результат для пользователя, то есть расписание занятий на определённый день или неделю. Функции View вызываются после того как Controller определит, что пользователь хотел в сообщении. Его функции:

– get_timetable_teacher(html, type='текущая неделя') – функция для формирования расписания преподавателя на неделю. На вход функция получает

html-текст страницы с расписанием и по нему ищет необходимые составляющие для расписания. Также получает тип недели: четная или нечетная. По умолчанию формируется расписание на текущую неделю. Возвращает текст с расписанием;

– get_timetable_week(html, type='текущая неделя') – аналогично функции get_timetable_teacher() только для студентов;

– get_timetable_day(html, day) – функция для формирования расписания на определённый день недели. Получает html-текст страницы с расписанием и день недели, который нужно сформировать. Возвращает текст с расписанием;

– get_timetable_today(html) – функция для формирования расписания для команды /today. На вход получает html-текст страницы с расписанием. Затем определяет текущий день недели. Возвращает текст с расписанием на этот день;

– get_timetable_tomorrow(html) - функция для формирования расписания для команды /tomorrow. Работает по аналогии с get_timetable_today();

– get_subscribers_timetable() – функция формирует и возвращает словарь всех подписчиков на ежедневную рассылку расписания. Словарь содержит ключ с chat_id пользователя и значением текст расписания;

– subscription(chat_id) – функция реализует процесс подписки или отписки на ежедневную рассылку расписания. На вход получает chat_id пользователя. Возвращает сообщение о том успешно подписался пользователь или отписался;

– update_table_of_urls() – функция для команды /update_table. Недоступна обычным пользователям. Воспользоваться ему может только администратор бота. Функция удаляет данные из таблицы с номерами групп и их адресами и заново заполняет новыми данными.

3.1.3 Модуль Model, содержит функции, которые непосредственно обращаются к данным в таблицах и возвращают результат в соответствии с sql-запросом. Его функции:

– get_all_subscribers() – функция возвращает словарь где первым значением является chat_id, а вторым соответствующий номер группы записанный в таблице;

– get_current_subscription(chat_id) – функция принимает chat_id пользователя и возвращает его значение подписки, либо true, либо false;

– subscription_on(chat_id) – функция принимает chat_id и данному пользователю устанавливает его subscription в значение True;

– subscription_off(chat_id) – функция принимает chat_id и данному пользователю устанавливает его subscription в значение False;

– clear_table_of_urls() – функция удаляет данные у таблицу urls_of_group;

– create_row_table_of_urls(d) – функция на вход получает словарь ключом которого являются номера групп, а значениями адреса их расписания. Затем функция заполняет таблицу urls_of_group;

– get_part_of_url(number_of_group) – функция на вход принимает номер группы, по которому в таблице urls_of_group будет искать адрес на расписание этой группы.

3.2 Хранение данных

Для использования команд, которые можно создать для бота через BotFather в Telegram, понадобилось вводить базу данных, чтобы хранить chat_id пользователя и его номер группы, который он задаст.

Для хранения данных была выбрана СУБД Postgres [14], так как на Heroku имеется расширение для быстрого и простого подключения базы данных для проекта – Heroku Postgres, а также оно совместимо с языком Python. Так у расширения есть бесплатная подписка Hobby Dev, которая позволяет бесплатно хранить суммарно 10000 строк во всех таблицах проекта. PostreSQL подключается через библиотеку psycopg2 [15].

Для хранения данных о пользователях была создана таблица users (рисунок 3.2). Таблица имеет пять полей:

– id (integer) – primary key, уникальное значение для каждой записи. На самом деле chat_id тоже является уникальным и его можно было бы использовать как уникальное поле для каждой строки, но лучше перестраховаться;

– chat_id (integer) – уникальный номер между пользователем и ботом, по которому отправляются все сообщения от бота;

– last_message (text) – последнее сообщение от пользователя. Сохраняется для того, чтобы отслеживать команду /registration, которая записывает группу пользователя;

– number_of_group (text) – номер группы пользователя. По данному полю происходит поиск расписания группы пользователя;

– subscription (boolean) – поле, которое хранит данные о том подписан пользователь на ежедневную рассылку или нет.

	id	chat_id	last_message	number_of_group	subscription
1	3	519358553	варламова о н	вц18-01вп	☐
2	4	465303626	/subscription	ки18-02/1б	☑
3	19	601618396	/update_table	сб18-11	☑
4	1	380622052	/tomorrow	ки15-17б	☐
5	18	336007117	вц18-01асу	сб18-11	☑
6	5	194642602	/subscription	ки17-17б2	☐
7	7	335965091	/today	ки18-09б	☐
8	6	9208325	ки17-17	<null>	☐
9	2	356928543	/	ки15-17б	☑

Рисунок 3.2 – Таблица пользователей

Также была создана таблица для хранения названий групп и ссылок на их расписание на сайте СФУ (рисунок 3.3). Таблица urls_of_group имеет четыре поля данных:

– id (integer) – primary key, уникальное значение для каждой записи. На самом деле number_of_group тоже является уникальным и его можно было бы использовать как уникальное поле для каждой строки, но лучше перестраховаться;

– number_of_group (text) – номер группы;

– number_of_group_en (text) – номер группы пользователя переведенный на латиницу. Так как на сервисе СФУ е-курсы используется для входа логин на латинице, то и у некоторых студентов могла сложиться ассоциация с тем, что номер группы надо писать на латинице;

– part_of_url (text) – ссылка на расписание группы.

	id	number_of_group	number_of_group_en	part_of_url
1	21083	вц18-01асу1	vts18-01asu1	?group=%D0%92%D0%A618-01%D0%90%...
2	21084	вц18-01асу2	vts18-01asu2	?group=%D0%92%D0%A618-01%D0%90%...
3	21085	вц18-01на1	vts18-01na1	?group=%D0%92%D0%A618-01%D0%9D%...
4	21086	вц18-01на2	vts18-01na2	?group=%D0%92%D0%A618-01%D0%9D%...
5	21087	вц18-01ртв1	vts18-01rtv1	?group=%D0%92%D0%A618-01%D0%A0%...
6	21088	вц18-01ртв2	vts18-01rtv2	?group=%D0%92%D0%A618-01%D0%A0%...
7	21089	вц18-02асу1	vts18-02asu1	?group=%D0%92%D0%A618-02%D0%90%...
8	21090	вц18-02асу2	vts18-02asu2	?group=%D0%92%D0%A618-02%D0%90%...

Рисунок 3.3 – Таблица номеров групп и ссылок на их расписание

Для создания базы данных и таблиц использовался инструмент DataGrip от компании JetBrains. Позволяет настраивать, как и локальных пользователей для управления базами данных, так и подключаться к удаленным базам на сторонних серверах. DataGrip позволяет создать различные источники подключения: PostgreSQL, MySQL, SQLite и множество других популярных и не очень.

3.3 Алгоритмы

3.3.1 Алгоритм обработки пользовательского сообщения (рисунок 3.4).

Изначально сообщение пользователя в виде post-запроса попадает в функцию index() с декоратором @app.route. Оттуда из post-запроса выбираются данные: chat_id пользователя и текст его сообщения, которые отправляются в функцию user_massages_handler(chat_id, message).

Так как сообщения пользователя можно подстроить под несколько шаблонов: это или команда, которая начинается на символ слэш, или номер группы, которые всегда начинается с двух букв, затем идет две цифры, а затем

тире. Благодаря этому можно ввести регулярные выражения, тем самым быстро обрабатывая верные запросы, и отбрасывая неверные, соответствующим сообщением о том, что данная команда не существует.

После того как сообщение пользователя обработалось вызывается соответствующая функция, которая отвечает за недельное расписание, или расписание сегодняшнего дня или команда и т.д. Которая в свою очередь собирает данные html с советующей группой пользователя с сайта СФУ, затем выбирает из этих данных необходимые и преобразует в текст, который будет отправлен пользователю.

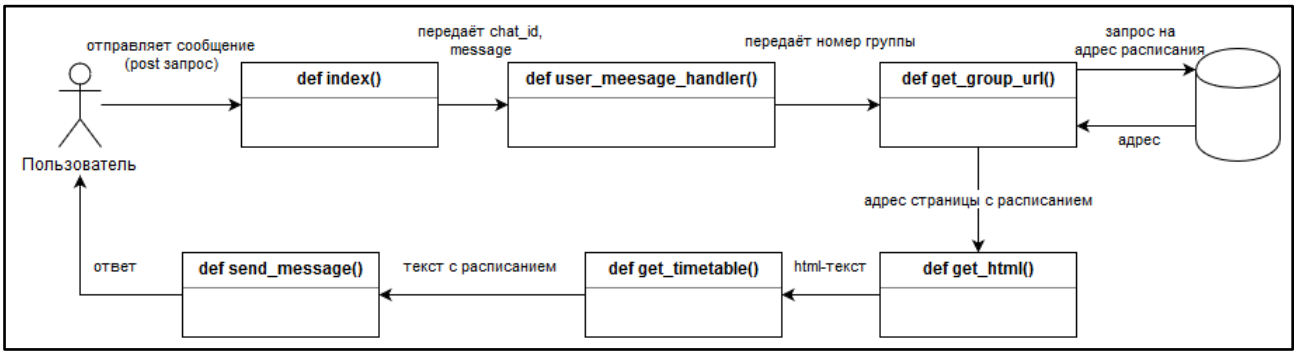

Рисунок 3.4 – Алгоритм обработки пользовательского сообщения

3.3.2 Алгоритм обновления таблицы с группами и ссылками на них (рисунок 3.5).

У бота есть команда /update_table, которая не доступна другим пользователям, кроме админа. На сервисе Heroku, где размещен бот, задано ключевое слово admin_id со значением chat_id пользователя, который владеет данным ботом. Админ отправляет данную команду боту. Бот парсит страницу с расписанием с сайта СФУ. Затем извлекает оттуда названия групп и ссылки на них. Названия групп приводятся к общему шаблону, а также переводятся на латиницу. Затем заполняется таблица в базе данных. Старые данные удаляются.

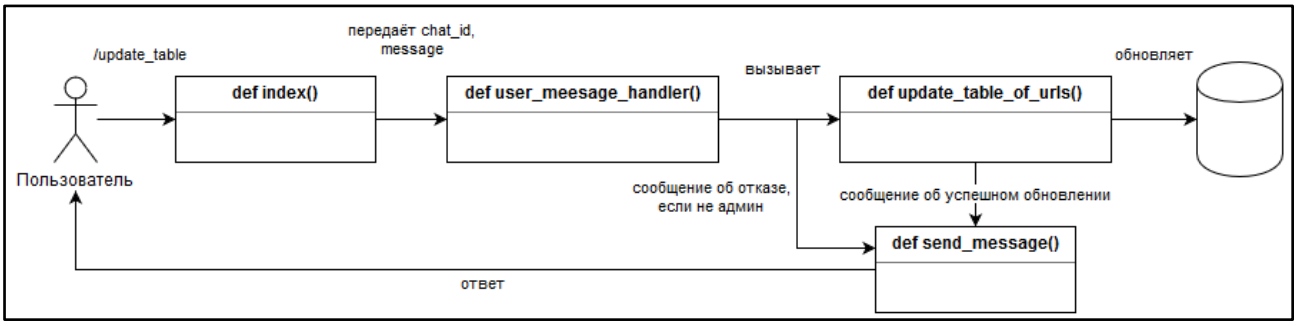

Рисунок 3.5 – Алгоритм обновления таблицы с группами

3.3.3 Алгоритм регистрации (рисунок 3.6).

Для того, чтобы воспользоваться командами боту нужно заранее знать какой номер группы у студента. Для этого последнему необходимо пройти процедуру регистрации. Так пользователь отправляет команду /registration, бот записывает последнее сообщение пользователя в таблицу, а пользователю отправляет просьбу написать номер группы. Если пользователь отправил номер группы, а последнее сообщение содержит «/registration», тогда бот пытается найти данную группу в таблице urls_of_group. Если бот успешно находит группу, то записывает ее в таблице users в поле number_of_group, в противном случае пользователю приходит сообщение об ошибке.

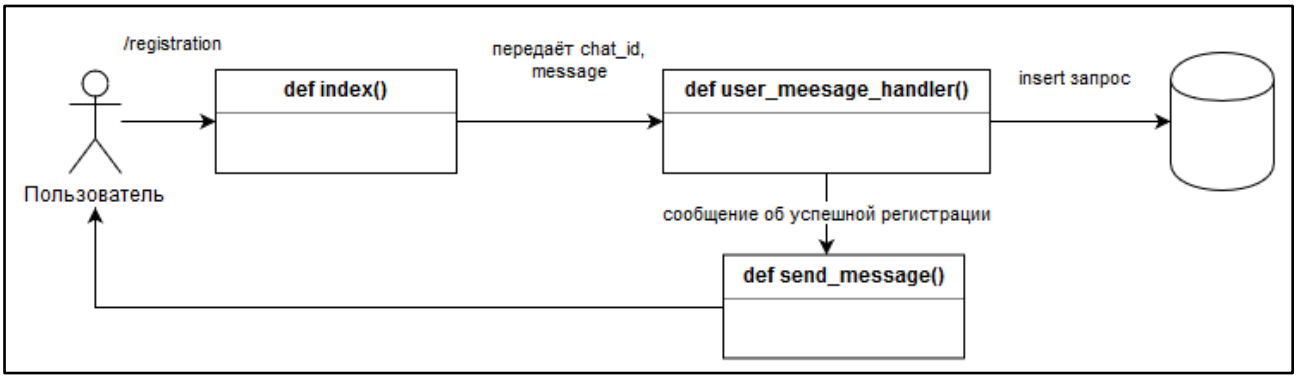

Рисунок 3.6 – Алгоритм регистрации

3.3.4 Алгоритм рассылки ежедневного расписания.

После того как пользователь зарегистрируется. Он может подписаться на ежедневное оповещение о расписании на завтрашний день (рисунок 3.7). Для

этого надо отправить команду /subscription, бот анализирует сообщение и проверит поле subscription в таблице users. Если у пользователя данное поле находится в значении false, бот заменит его на true и пользователь начнет получать оповещение каждый день вечером. Если поле subscription находится в значении true, тогда бот заменит его на false и пользователь перестанет получать оповещения.

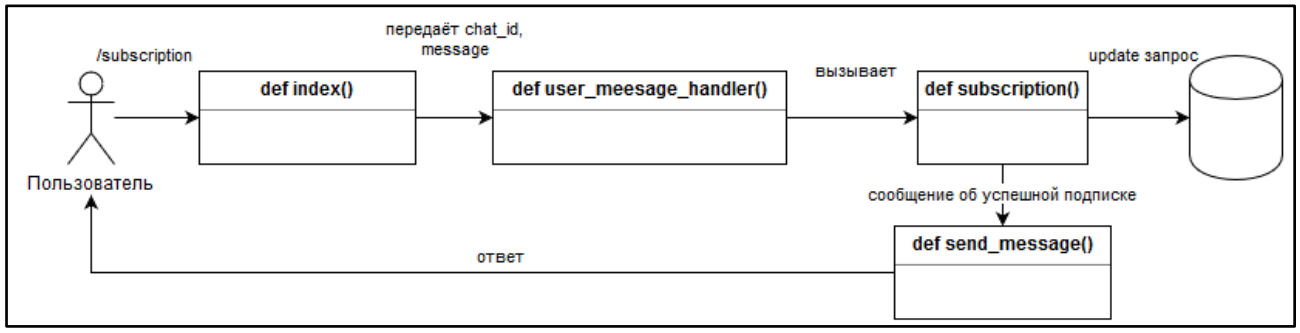

Рисунок 3.7 – Алгоритм подписки

Для ежедневного оповещения были использованы библиотеки threading и schedule. Библиотека threading позволяет создать параллельный поток основному, который отвечает за прием сообщений пользователя. Библиотека schedule позволяет создать объект своего класса, который будет отвечать за исполнение функции в определенный момент времени [16]. Поток отвечает за исполнение функции def_schedule_run(), которая содержит цикл, внутри которого запущен объект schedule (рисунок 3.8).

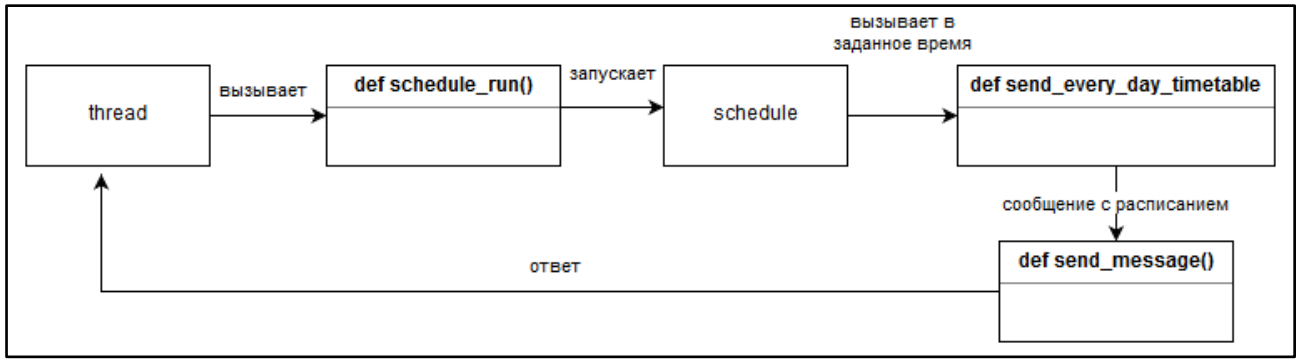

Рисунок 3.8 – Алгоритм ежедневного оповещения

4 Демонстрация работы

Далее в главе будут рассмотрены два варианта обращения к боту для получения расписания: через команды или через полное написание запроса.

Каждое расписание имеет свой шаблон.

Так, например, расписание на «сегодня», «завтра», «определённый день» недели имеет следующую структуру:

Первая строка: группа {номер}

Вторая строка: {день недели}/{тип недели (первая, вторая)}

Третья строка: {номер ленты} {время начала и конец пары}

Четвертая строка: {название предмета} и {тип предмета}

Пятая строка: {имя преподавателя}

Шестая строка: Каб:{номер аудитории}

Между предметами проставляется пустая строка для удобочитаемости.

Расписание по подписке также содержит строку с датой для того, чтобы пользователь понимал на какой момент времени это расписание.

Расписание на «неделю», «четную неделю» и «нечетную неделю» не разбивается на строки, только дни между собой отделяются пустой строкой, так как получается очень длинное полотно текста и его неудобно читать.

4.1 Получение расписания через полное написание запроса

Если написать боту номер группы и через пробел ключевое слово «сегодня», то пользователь получит расписание на сегодняшний день (рисунок 4.1).

Если написать боту номер группы и через пробел ключевое слово «завтра», то пользователь получит расписание на завтрашний день (рисунок 4.2).

Если написать боту номер группы и через пробел сокращенное название дня недели, то пользователь получит расписание на этот день текущей недели (рисунок 4.3).

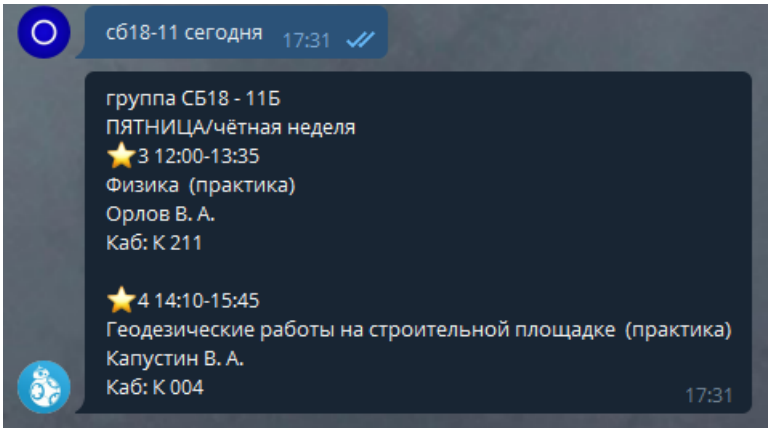

Рисунок 4.1 – Расписание на сегодня

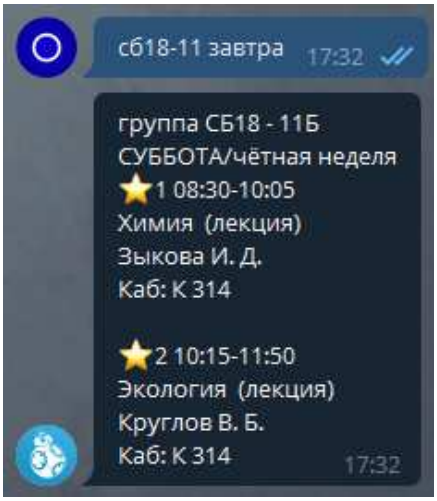

Рисунок 4.2 – Расписание на завтра

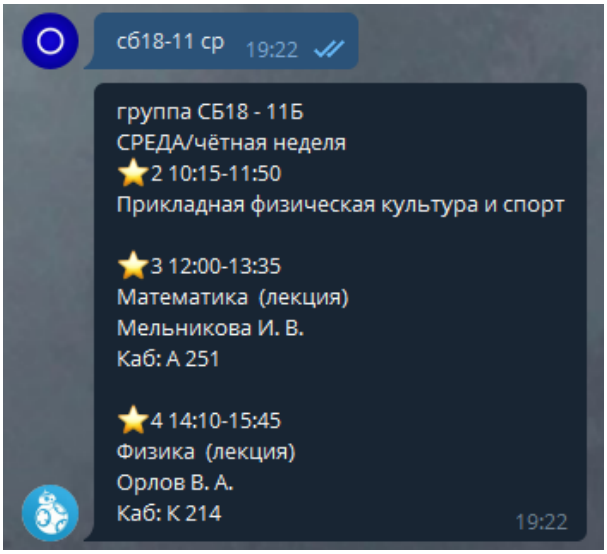

Рисунок 4.3 – Расписание на определённый день

Если написать боту только номер группы, то пользователь получит расписание на текущую неделю для этой группы (рисунок 4.4).

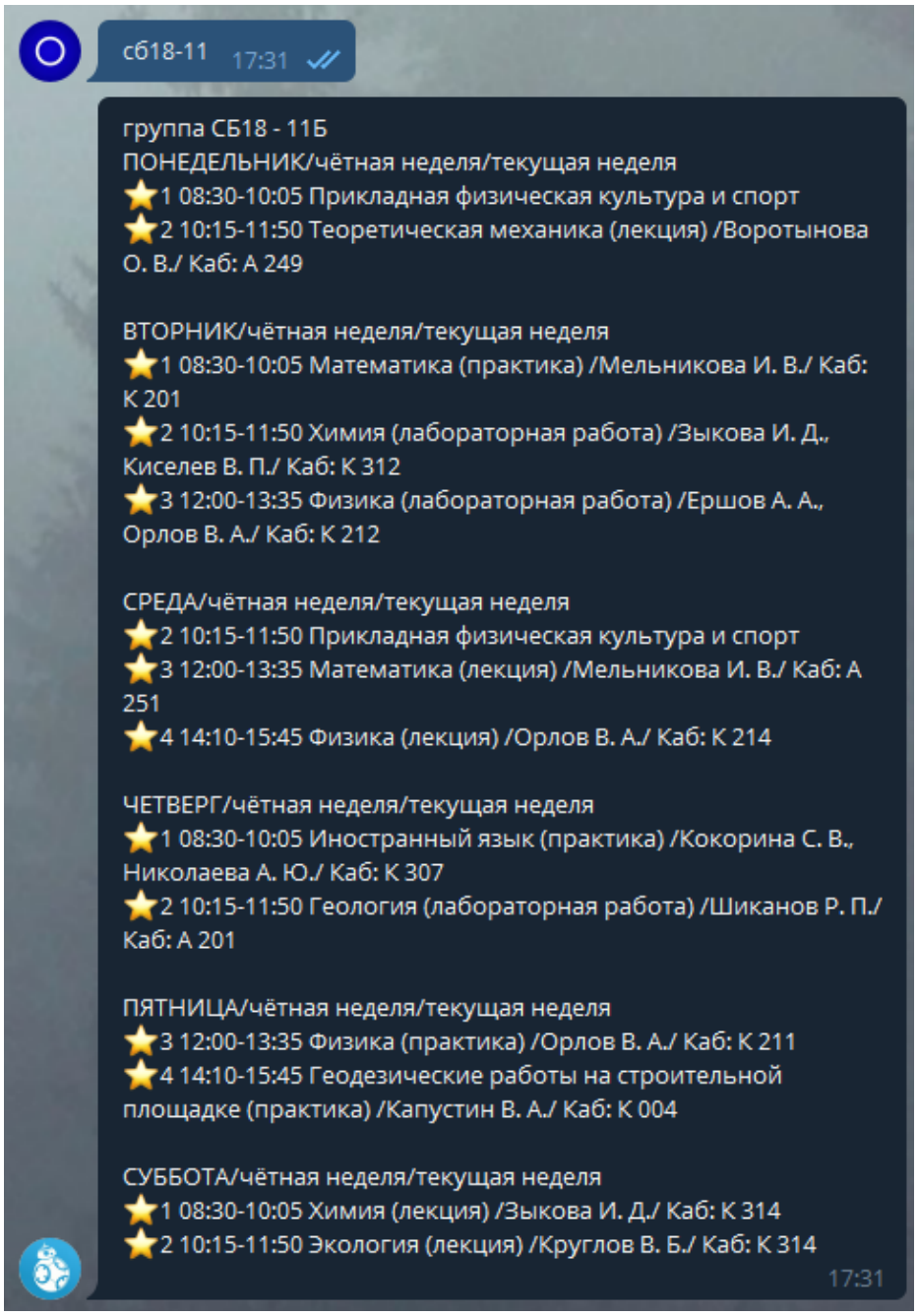

Рисунок 4.4 – Расписание на неделю

Если боту написать фамилию преподавателя и его инициалы, то пользователь получит расписание преподавателя на текущую неделю (рисунок 4.5).

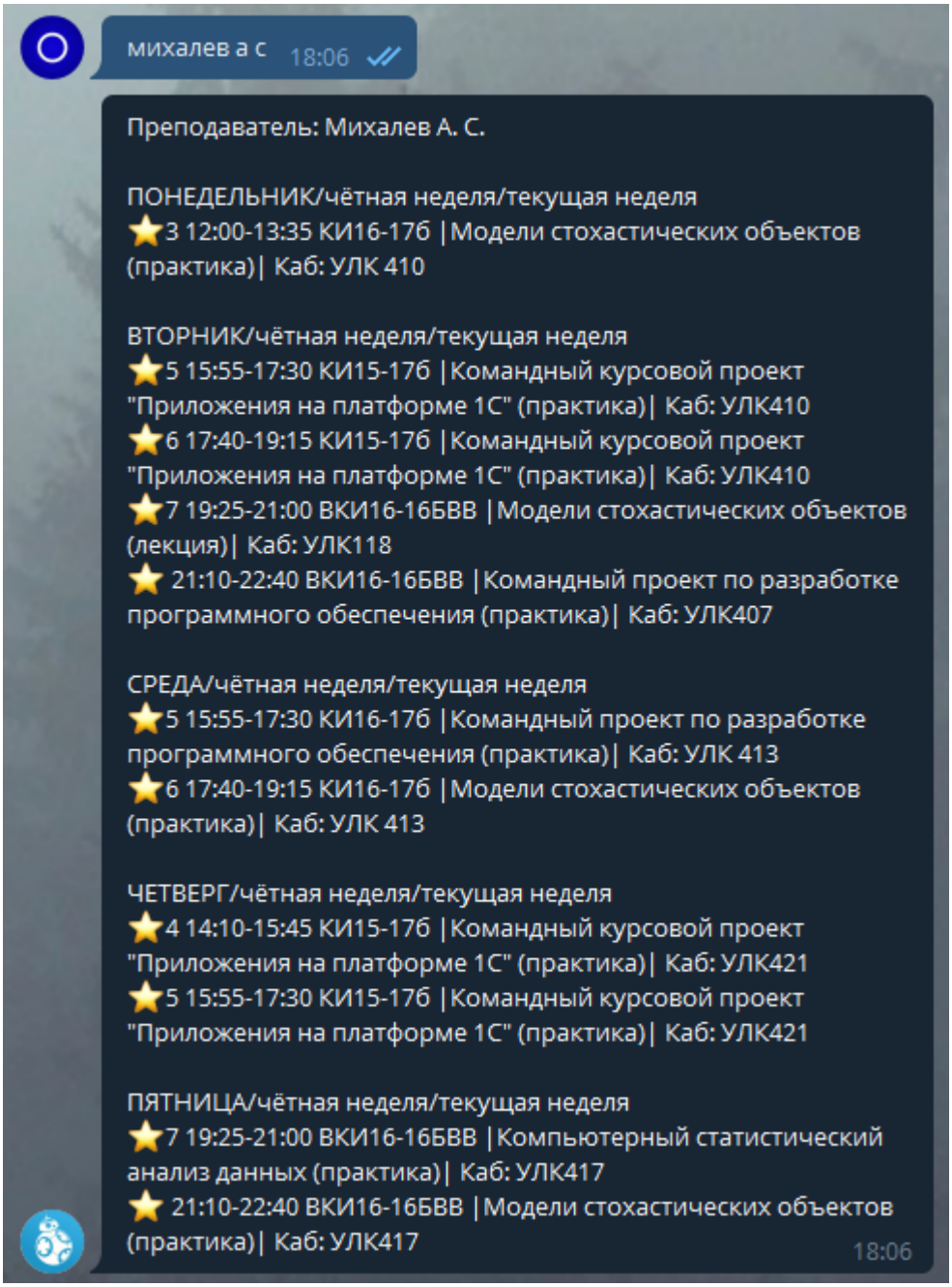

Рисунок 4.5 – Расписание преподавателя

Если пользователь ввел номер несуществующей группы (рисунок 4.6) или инициалы несуществующего преподавателя (рисунок 4.7), то получит соответствующее сообщение об отсутствии расписания группы или преподавателя.

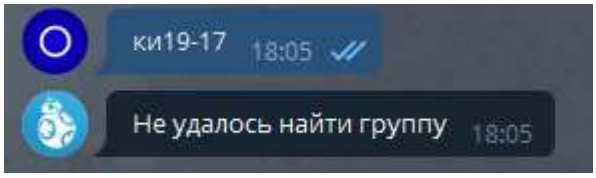

Рисунок 4.6 – Пользователь ввел несуществующую группу

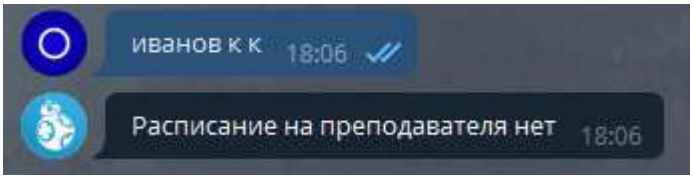

Рисунок 4.7 – Пользователь ввел несуществующего преподавателя

Если пользователь ввел запрос, не соответствующий ни одному регулярному выражению, тогда он получает сообщение, что его запрос был составлен неправильно (рисунок 4.8).

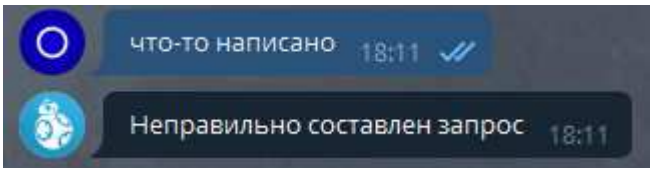

Рисунок 4.8 – Пользователь ввел неправильный запрос

4.2 Получение расписания через команды

Для того чтобы пользователь мог пользоваться командами ему нужно отправить боту команду /registration, бот попросит ввести пользователя номер группы и запишет его в базу данных (рисунок 4.9).

Затем пользователю станет доступен ряд команд:

– /help – выводит пользователю инструкцию по использованию бота (рисунок 4.10);

– /week – выдает пользователю расписание на текущую неделю (рисунок 4.11);

– /today – выдает пользователю расписание на сегодняшний день (рисунок 4.12);

– /tomorrow – выдает пользователю расписание на завтрашний день (рисунок 4.13);

– /subscription – команда (рисунок 4.14) для включения ежедневного оповещения о завтрашнем расписании (рисунок 4.15);

– /update_table – команда для обновления таблицы с адресами групп (рисунок 4.16).

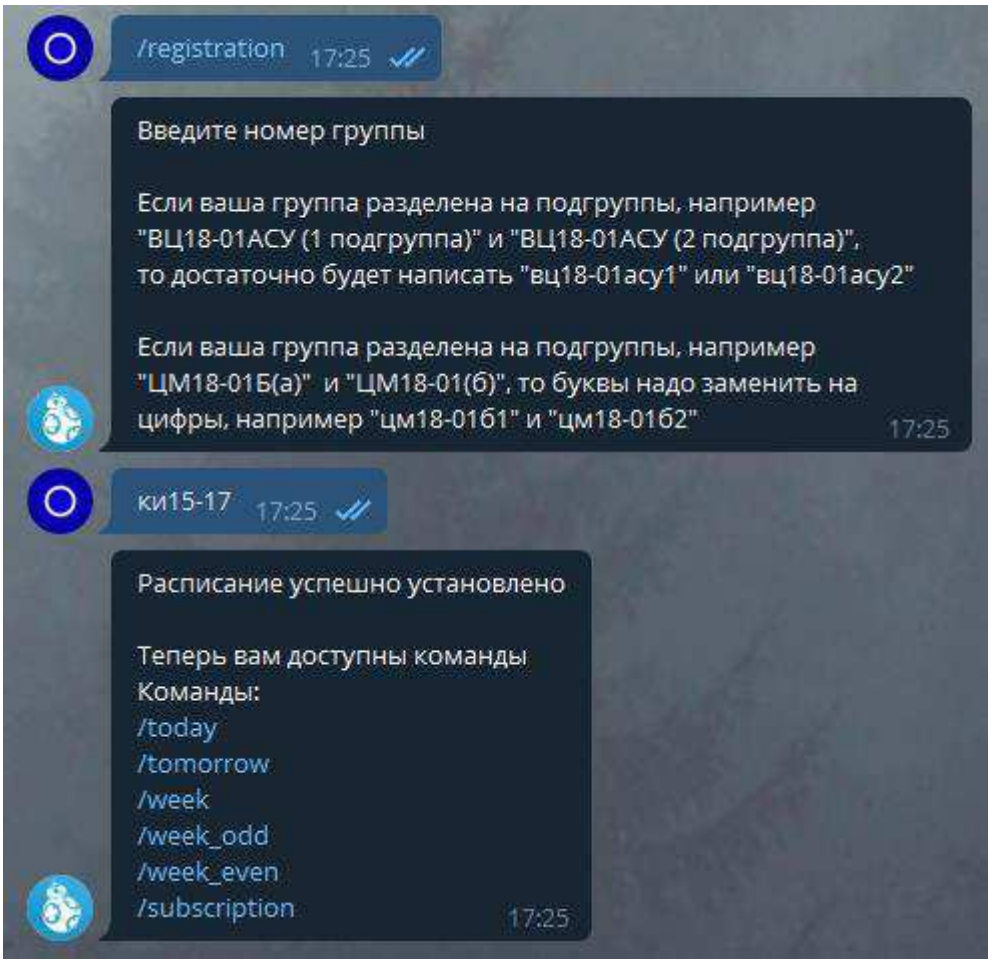

Рисунок 4.9 – Процесс регистрации

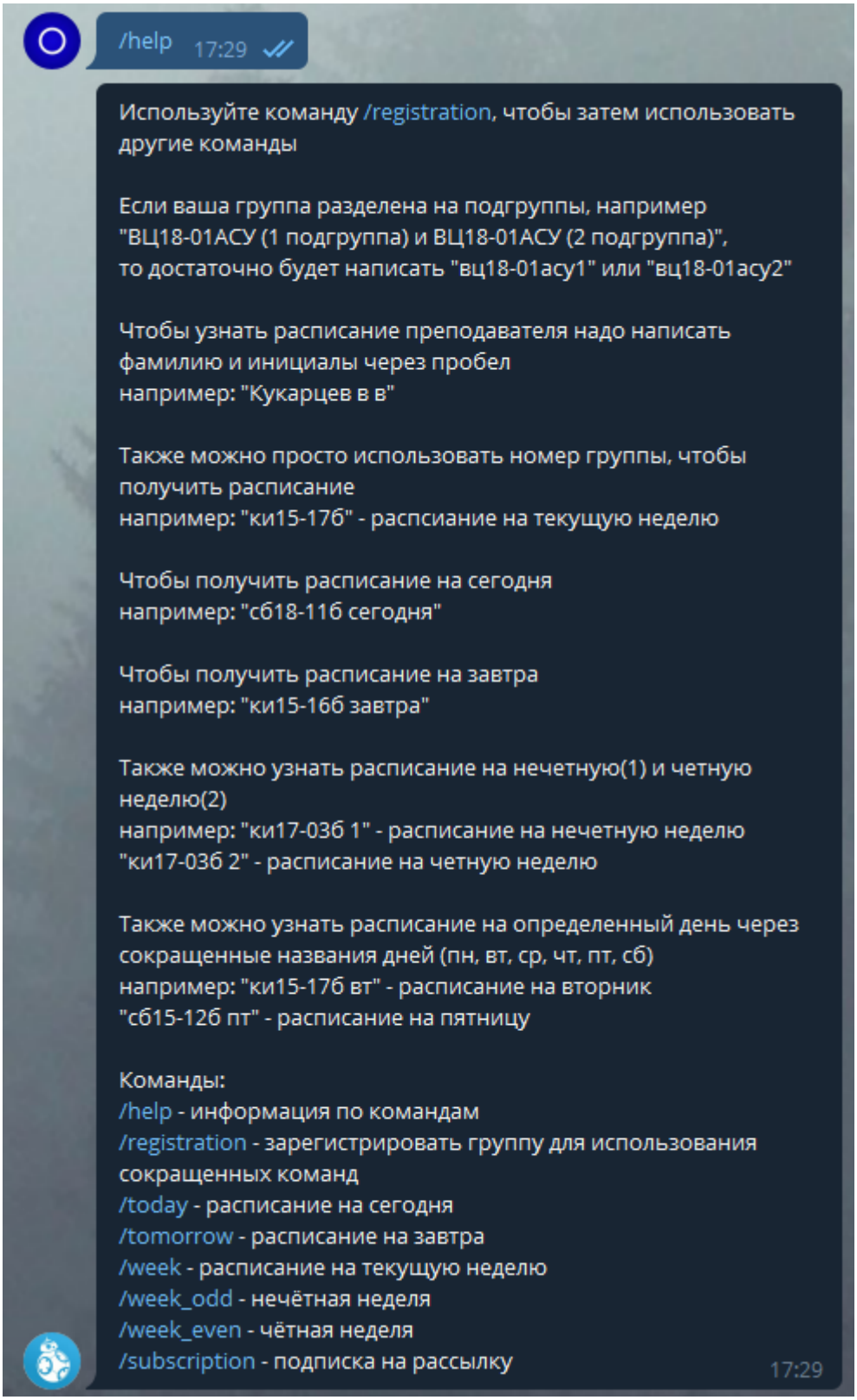

Рисунок 4.10 – Инструкция для пользователя

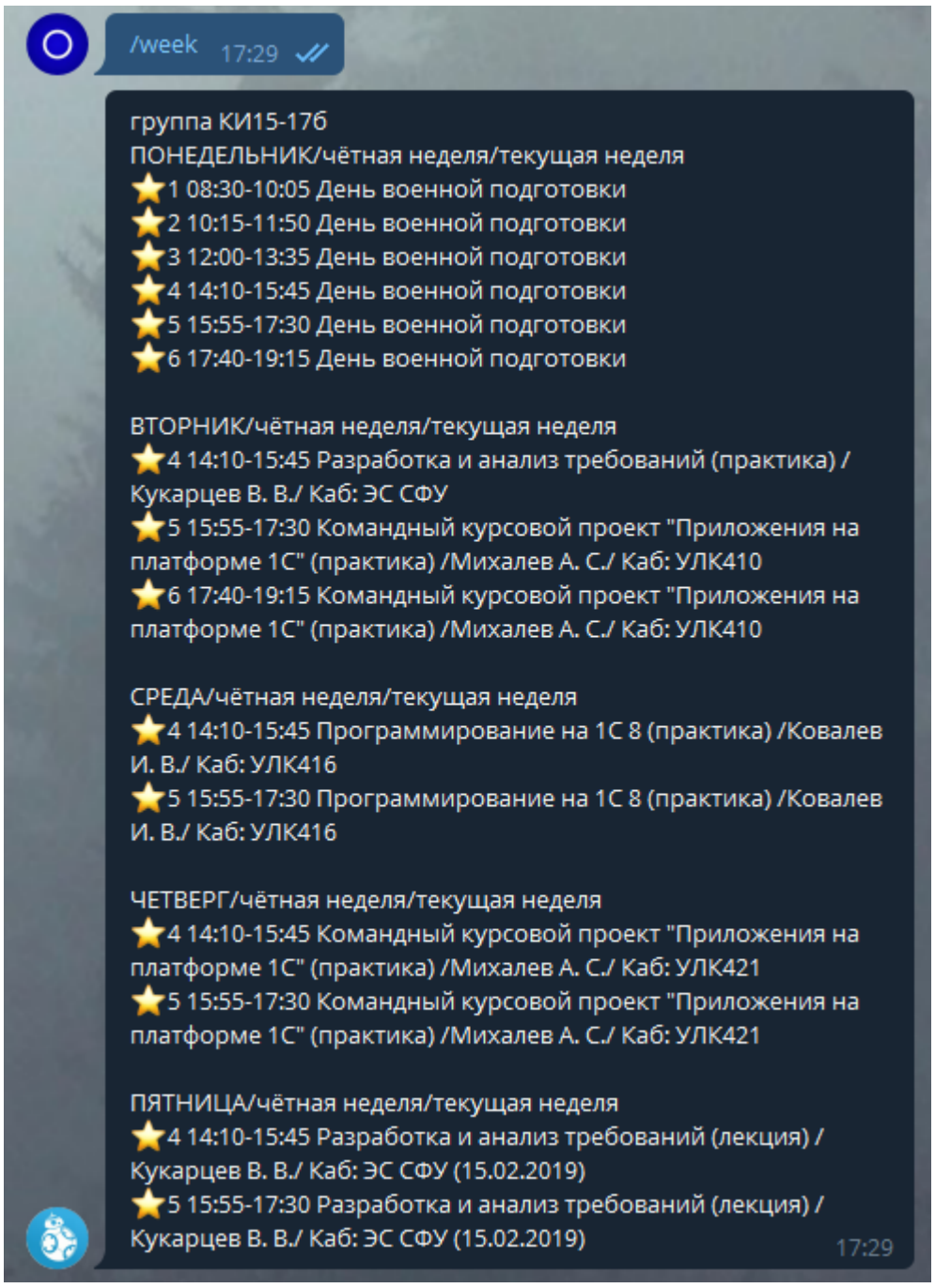

Рисунок 4.11 – Расписание на текущую неделю

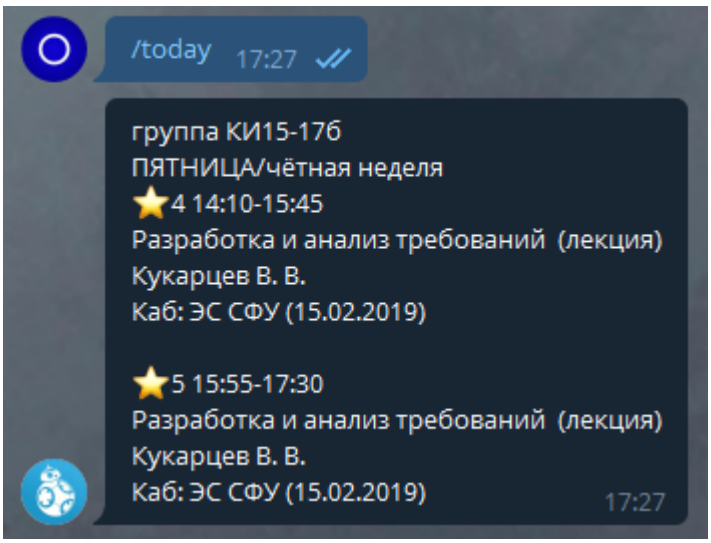

Рисунок 4.12 – Расписание на сегодняшний день

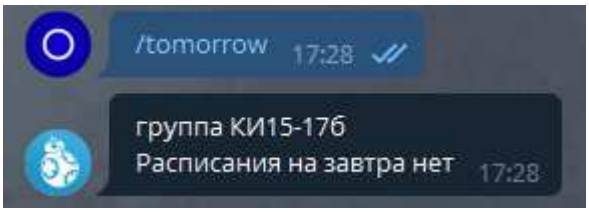

Рисунок 4.13 – Расписание на завтрашний день

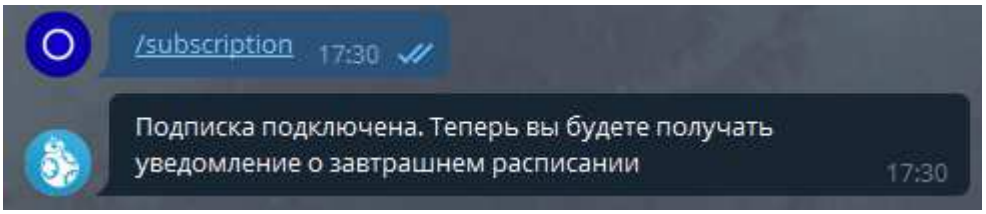

Рисунок 4.14 – Процесс включения уведомлений

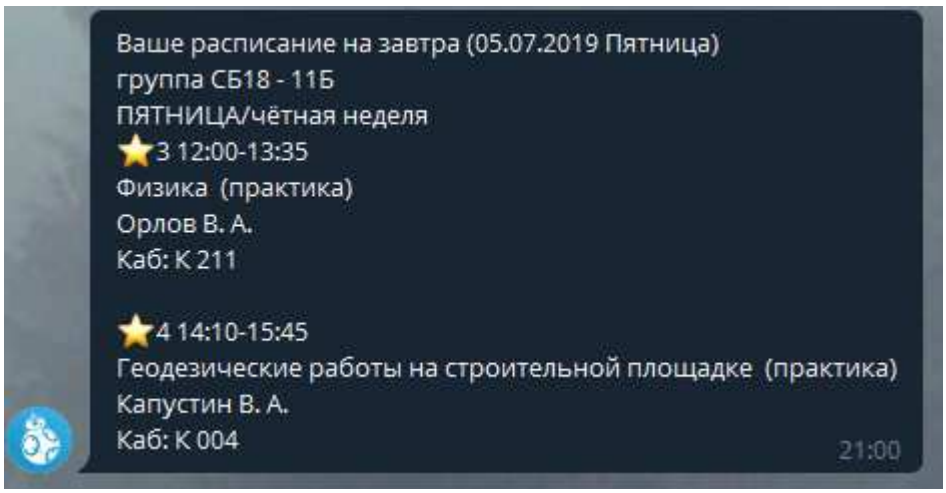

Рисунок 4.15 – Ежедневное уведомление

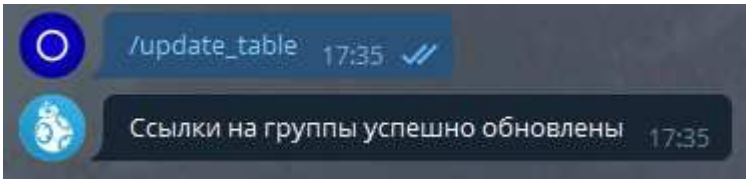

Рисунок 4.16 – Обновление списка групп и адресов

Если пользователь ввел некорректную команду, то он получает соответствующее сообщение (рисунок 4.17).

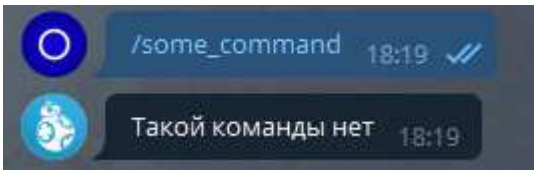

Рисунок 4.17 – Некорректная команда

ЗАКЛЮЧЕНИЕ

В данной выпускной квалификационной работе был создан и опсан чат-бот для мессенджера Telegram, который предоставляет студентам СФУ актуальное для них расписание занятий.

Бот разработан на языке Python. Для хранения данных использована СУБД PostgreSQL. Для хостинга использовался сервис Heroku. Данные берутся с главного сайта СФУ с соответствующего раздела с расписанием. Парсинг данных реализован с помощью python-библиотеки BeautifulSoup4. Исходный код размещен на сервисе GitHub. Мысли и идеи, а также отслеживание их выполнения было записано на доску в Trello. Также в процессе разработки было найдено множество ответов на вопросы на тематических сайтах, посвященных IT-разработке, таких как Тостер и Stack Overflow, и других форумах.

Бот может отвечать на команды или работать как поисковик. Чтобы бот воспринимал встроенные команды от пользователя, последнему нужно пройти простую регистрацию – написать свой номер группы после команды /registration. Бот запишет пользователя и его группу в базу данных. Чтобы бот работал как поисковик, нужно просто написать любой номер группы и тем самым получить расписание занятий на неделю. Если добавить к номеру группы день недели, то можно получить расписание на этот день. Также, если написать фамилию и инициалы преподавателя, то можно получить его расписание на текущую неделю.

Данный чат-бот станет отличным дополнением для студента в ходе его обучения. А возможно, и заменой существующим аналогам.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Безопасность в интернете: программы-боты // Блог Яндекса [Электронный ресурс]. – Режим доступа: https://yandex.ru/blog/company/85243 (дата обращения 17.06.2019).

2. 80% of businesses want chatbots by 2020 // Business Insider Intelligence [Электронный ресурс]. – Режим доступа: https://www.businessinsider.com/80-of-businesses-want-chatbots-by-2020-2016-12 (дата обращения 17.06.2019).

3. Исследование аудитории Telegram 2019 // Telegram Analytics [Электронный ресурс]. – Режим доступа: https://tgstat.ru/research (дата обращения 17.06.2019).

4. Interactive: The Top Programming Languages 2018 // IEEE Spectrum [Электронный ресурс]. – Режим доступа: https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2018 (дата обращения 17.06.2019).

5. TIOBE Index for June 2019 // TIOBE [Электронный ресурс]. – Режим доступа: https://www.tiobe.com/tiobe-index (дата обращения 17.06.2019).

6. Flask Routing // Junxian’s Documentation [Электронный ресурс]. – Режим доступа: https://junxiandoc.readthedocs.io/en/latest/docs/flask/flask_routing.html (дата обращения 17.06.2019).

7. Маленькие хитрости с Heroku // JavaRush [Электронный ресурс]. – Режим доступа: https://javarush.ru/groups/posts/1987-malenjhkie-khitrosti-s-heroku (дата обращения 17.06.2019).

8. Heroku Postgres // Heroku Dev Center [Электронный ресурс]. – Режим доступа: https://devcenter.heroku.com/articles/heroku-postgresql (дата обращения 17.06.2019).

9. WhatsApp was hacked and attackers installed spyware on people's phones // Business Insider [Электронный ресурс]. – Режим доступа: https://www.businessinsider.com/whatsapp-hacked-attackers-installed-spyware-2019-5?r=US&IR=T (дата обращения 17.06.2019).

10. WhatsApp is broken, really broken // fileperms.org [Электронный ресурс]. – Режим доступа: https://web.archive.org/web/20150108072201/http://fileperms.org/whatsapp-is-broken-really-broken.html (дата обращения 17.06.2019).

11. Руководство «ВКонтакте»: «Мы уже несколько лет сотрудничаем с ФСБ // Новая газета [Электронный ресурс]. – Режим доступа: https://www.novayagazeta.ru/articles/2013/03/27/54100-rukovodstvo-171-vkontakte-187-171-my-uzhe-neskolko-let-sotrudnichaem-s-fsb-i-otdelom-171-k-187-mvd-operativno-vydavaya-informatsiyu-o-tysyachah-polzovateley-nashey-seti-187 (дата обращения 17.06.2019).

12. Аудитория самых популярных мессенджеров в России на начало 2018 года: Telegram в тройке лидеров // vc.ru [Электронный ресурс]. – Режим доступа: https://vc.ru/flood/36321-auditoriya-samyh-populyarnyh-messendzherov-v-rossii-na-nachalo-2018-goda-telegram-v-troyke-liderov (дата обращения 10.06.2019).

13. Обобщенный Model-View-Controller // RSDN [Электронный ресурс]. – Режим доступа: http://rsdn.org/article/patterns/generic-mvc.xml (дата обращения 10.06.2019).

14. Документация к PostgreSQL 10.1 // postgrespro.ru [Электронный ресурс]. – Режим доступа: http://repo.postgrespro.ru/doc/pgsql/10.1/ru/postgres-A4-fop.pdf (дата обращения 10.06.2019).

15. Psycopg2 Tutorial // PostgreSQL Wiki [Электронный ресурс]. – Режим доступа: https://wiki.postgresql.org/wiki/Psycopg2_Tutorial (дата обращения 10.06.2019).

16. Schedule // schedule.readthedocs.io [Электронный ресурс]. – Режим доступа: https://schedule.readthedocs.io/en/stable/ (дата обращения 10.06.2019).

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра «Информатика»

УТВЕРЖДАЮ

Заведующий кафедрой

_____А.С. Кузнецов

« 09 » июля 2019 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.04 - Программная инженерия

Разработка программного бота формирования расписания образовательного процесса в учебном заведении

Руководитель _____ 08.07.19 старший преподаватель А. С. Михалёв

Выпускник _____ 08.07.19 А. М. Отто

Консультант _____ 08.07.19 доцент, канд.техн.наук А. С. Кузнецов

Нормконтролер _____ 08.07.19 доцент, канд.техн.наук О. А. Антамошкин

Красноярск 2019