

**НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ УКРАИНЫ
«КИЕВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
имени ИГОРЯ СИКОРСКОГО»
Факультет информатики и вычислительной техники
Кафедра автоматки и управления в технических системах**

«К защите допущено»

Заведующий кафедрой

_____ О.И. Ролик

«__» _____ 2019 г.

Дипломный проект

на соискание степени бакалавра

по направлению подготовки 6.050103 «Программная инженерия»

на тему: «Телеграмм-бот построения маршрута с использованием общественного транспорта города Киева»

Выполнил:

студент IV курса, группы ИТ-51

Шородок Владимир Владимирович _____

Руководитель:

Доцент Писаренко Андрей Владимирович _____

Рецензент: _____

Подтверждаю, что в этом дипломном проекте нет заимствований из работ других авторов без соответствующих ссылок.

Студент _____

Киев – 2019 год

Национальный технический университет Украины
«Киевский политехнический институт имени Игоря Сикорского»
Факультет информатики и вычислительной техники
Кафедра автоматизации и управления в технических системах

Уровень высшего образования – первое (бакалаврское)

Направление подготовки – 6.050103 «Программная инженерия»

УТВЕРЖДАЮ

Заведующий кафедрой

_____ О.И. Ролик

«__» _____ 2019 г.

ЗАДАНИЕ

на дипломный проект студенту

Шородко Владимиру Владимировичу

1. Тема проекта «Телеграмм-бот построения маршрута с использованием общественного транспорта города Киева» руководитель проекта доцент кафедры АУТС Писаренко Андрей Владимирович, утвержденные приказом по университету от «__» _____ 2019 г. № _____

2. Срок представления студентом проекта _____

3. Исходные данные к проекту

Операционная система Windows, языки программирования Python и Javascript, среда программирования Microsoft Visual Studio, целевая платформа Telegram, выбранная для разработки технологии – Telebot, CherryPy, Requests, СУБД – MongoDB.

4. Содержание пояснительной записки

1. Введение 2. Проектирование 3. Анализ требований 4. Анализ предметной области 5. Обзор существующих решений 6. Разработка телеграмм-бота. 7. База данных – MongoDB 8. Telegram 9. Тестирование системы 10. Описание своей реализации. 11. выводы

Приложения: _____

5. Код программы _____

6. Список графического материала

UML Диаграмма вариантов использования, UML Диаграмма активности телеграмм-бота, UML Диаграмма классов айроки сообщения, UML Диаграмма последовательности обработки пользовательского запроса.

7. Дата выдачи задания _____

Календарный план

№ з/п	Название этапов выполнения дипломного проекта	Срок исполнения этапов проекта	Примечание
1	Выбор тематического направления согласование темы дипломного проекта да	22.02.2019	
2	Анализ теоретических материалов изучение предметной области да	15.04.2019	
3	Разработка технического задания, выбор методов и средств реализации задачи	24.04.2019	
4	Обзор существующих решений по тематике работы	27.04.2019	
5	Разработка структуры прототипа проектирование системы да	06.05.2019	
6	Реализация проекта	20.05.2019	
7	Настройка и проверка программы	23.05.2019	
8	Оформление пояснительной записки	03.06.2019	
9	Предзащита дипломного проекта	04.06.2019	
10	Доработка пояснительной записки и подготовка презентации	18.06.2019	
11	Защита дипломного проекта	20.06.2019	

Студент

Шеродок В.В.

Руководитель проекта

Писаренко А.В.

АННОТАЦИЯ

Шеродок В.В. Телеграмм-бот построения маршрута с использованием общественного транспорта города Киева КПИ им. Игоря Сикорского, Киев, 2019.

Объяснительная записка дипломного проекта содержит 15 таблиц, 1 приложение, 12 рисунков и 21 источник – всего 66 страниц.

Объект исследования: чат-бот постройки маршрута с использованием общественного транспорта города Киева.

Цель дипломного проекта: повышение удобства пользования общественным транспортом посредством разработки телеграмм-бота, строящего маршруты с использованием общественного транспорта г. Киева.

КЛЮЧЕВЫЕ СЛОВА: БОТ, ЧАТ-БОТ, РАБОТ, ТЕЛЕГРАМ-БОТ, PYTHON, МЕССЕНДЖЕР, TELEGRAM, МАРШРУТЫ, ОБЩЕСТВЕННЫЙ ТРАНСПОРТ.

SUMMARY

Shroodok VV Телеграмма по созданию коммуникации с помощью общественного транспорта в городе Киев Игорь Сикорский КPI, Kyiv, 2019.

Explanatory note of the diploma project contains 15 tables, 1 application, 12 figures and 21 sources - a total of 66 pages.

Object of research: chat-bot of construction of route using public transport of city of Kiev.

Существует диплома проект: увеличить способность использования общественного транспорта через развитие телеграмм, которые создают коммуникации с помощью общественного транспорта в Киеве.

KEYWORDS: BOT, CHAT-BOT, ROBOT, TELEGRAM-BOT, PYTHON, MESSENGER, TELEGRAM, ROADS, PUBLIC TRANSPORT.

Номер строки	Формат	Обозначение	Наименование	Кол. писем	№ экз.	Примечание
1			<u>Документация общая</u>			
2						
3			Вновь разработана			
4						
5	A4	ИТ51.320БАК.002 ПО	объяснительная записка	61		
6						
7	A3	ИТ51.320БАК.003 Д1	Диаграмма активности	1		
8						
9	A3	ИТ51.320БАК.004 Д2	Диаграмма вариантов использование	1		
10						
11	A3	ИТ51.320БАК.005 Д3	Диаграмма последовательностей	1		
12						
13	A3	ИТ51.320БАК.006 Д4	Диаграмма классов	1		
14						
15						
16						
17						
18						
19						
20						
21						
22						

				ИТ51.320БАК.001 ТП		
Изм.	Арк.	№ докум.	Подпись	Дата		
Разработ.		Шеродок В.В.			Лит.	Арк.
Проверь.		Писаренко А.В.				1
Рецензирование.					КПИ им. Игоря Сикорского ФИОТ, гр. ИТ-51	
Н. Контр.		Шинкевич М.К.				
Утверд.						
					Телеграмм-бот для построения маршрута с использованием общественного транспорта. Ведомость технического проекта	
					1	

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	2
ВВЕДЕНИЕ	4
1 ПРОЕКТИРОВАНИЕ.....	7
2 АНАЛИЗ ТРЕБОВАЛ.....	8
3 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	10
4 ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ	11
4.1. Приложение Eway.....	11
4.2. Synbot	12
5 РАЗРАБОТКА TELEGRAM-БОТА	14
5.1 Регистрация чат-бота для Telegram Bot API	14
5.2 Размещение чат-бота на удаленном сервере Heroku	15
5.3 Среда разработки.....	16
5.4 Язык программирования – Python	18
5.4.1 Модуль TeleBot	19
5.4.2 Модуль CherryPy	20
5.4.3 Модуль Requests	21
5.5 REST API, HTTP-запросы, JSON и череда сообщений	21
5.5.1 REST API.....	21
5.5.2 HTTP-запросы.....	22
5.5.3 Формат JSON.....	24
5.5.4 Платформа Heroku	27
5.5.5 Очередь сообщений	28
5.6 Telegram Bot API.....	29
6 БАЗА ДАННЫХ – MONGODB.....	46
7 TELEGRAM	50

					ИТ51.320БАК.001 ТП			
Изм.	Арк.	№ докум.	Подпись	Дата				
Разработ.		Шеродок В.В.			Телеграмм-бот для построения маршрута с использованием общественного транспорта. Ведомость технического проекта	Лит.	Арк.	Акрушев
Проверь.		Писаренко А.В.					2	61
Рецензирование.						КПИ им. Игоря Сикорского ФИОТ, гр. ИТ-51		
Н. Контр.		Шинкевич М.К.						
Утверд.								

8 ТЕСТИРОВАНИЕ СИСТЕМЫ.....	52
9 ОПИСАНИЕ СОБСТВЕННОЙ РЕАЛИЗАЦИИ.....	57
ВЫВОДЫ.....	59

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		3

ВВЕДЕНИЕ

Поскольку современный мир не стоит на месте, технологии стремительно развиваются и каждый день появляются новые тренды и идеи, многие имеют возможность повлиять на ИТ отрасль, которая в свою очередь влияет на другие отрасли, что в дальнейшем будет влиять на нашу повседневную жизнь. Если до недавнего времени популярны были приложения или компьютерные программы, то на данный момент лидерство занимают чат-боты, имеющие большие перспективы в разных сферах нашей жизни.

Для начала нужно определить, что представляет собой чат-бот. Чат-бот – это специализированное приложение, позволяющее пользователям взаимодействовать со сторонними сервисами, если существует такая необходимость и все это выполнено через известный интерфейс чата.

Чат-бот – это некоторый помощник, который общается с пользователями через сообщения и имеет множество определенных функций. То есть можно получить определенную информацию, написав чат-боту специальную команду, которую в свою очередь последний интерпретирует определенным образом. Так можно быстро переводить, комментировать, находить, тестировать, искать, обучать, транслировать, встраиваться в другие сервисы и платформы, взаимодействовать с датчиками и вещами, подключенными к Интернету.

AI (Artificial intelligence) в общем используется для сокращения издержек и повышения качества услуг. Наиболее популярными направлениями развития AI являются распознавание голоса и текстовое общение с помощью разных поисковых роботов. К примеру можно упомянуть обо всем известных всплывающих окошках "консультантов" на многих сайтах. Однако о качестве таких консультаций можно поспорить, но тенденция очевидна и от нее не собираются отступать. Ярким примером автоматизации можно назвать множество ботов для приема заявок на доставку еды, бронирование столиков в ресторанах, рассылки рекламы и многое другое.

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		4

Такие боты позволяют увеличить прибыль компаний, поскольку бот может обрабатывать заявки с большей скоростью, чем человек, и снизить нагрузку на рабочий персонал.

Каждый день миллионы людей пользуются общественным транспортом города Киева, часто возникает необходимость быстро узнать о ближайших остановках, какой транспорт останавливается на них, когда прибудет на остановку и построить оптимальные маршруты.

Также в Киеве, отрасль общественного транспорта развивается, на большинстве маршруток, трамваев и троллейбусов уже установлены GPS – маячки, которые дают возможность отслеживать местонахождение транспорта и рассчитывать время прибытия на ту или иную остановку.

Поэтому, для решения определенных потребностей, приведенных ранее, чтобы сохранить время людей и сделать планирование маршрута легким делом, было решено создать телеграмм-бот построения маршрутов с использованием общественного транспорта в Киеве на платформе Telegram.

Telegram – это приложение, разработанное на языке программирования C++, позволяющее обмениваться сообщениями и файлами большинства форматов. Мессенджер использует специально разработанную серверную часть с закрытым кодом, работающих на серверах Германии и США. Telegram имеет ряд преимуществ:

- а) конфиденциальность – все чаты зашифровываются, а сообщения уничтожаются через указанное время;
- б) скорость – скорость доставки сообщений выше, чем у аналогов; в) оптимизация – серверы Telegram расположены по всему миру, что повышает надежность;
- г) открытость – использование открытого протокола; д) отсутствие подписок и рекламы;
- е) отсутствие ограничений на размер сообщений и вложенных файлов.

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		5

Актуальность бакалаврской работы обусловлена высокой популярностью мессенджеров и средств автоматизации как чат-боты среди пользователей сети Интернет. Чат-боты позволяют упростить ежедневные рутинные задачи, такие как получение информации о погоде, пробках, последних новостях и другие. Главным достоинством классических приложений является возможность размещения всех возможностей на платформе одного мессенджера.

Была поставлена цель: создать телеграмм-бота, который сможет в соответствии с местонахождением пользователей, предоставить ему необходимую информацию о ближайших остановках, какой именно общественный транспорт останавливается на полученных остановках, при необходимости сообщать когда прибудет на остановку необходимый транспорт, строить оптимальные маршруты в соответствии с запросом пользователя .

Исходя из поставленных целей, были поставлены следующие задачи: а)

анализ выбранной предметной области;

б) сравнение имеющихся аналогов чат-ботов; в)

выбор технологий и среды разработки; г)

разработка чат-бота на платформе Telegram.

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		6

1 ПРОЕКТИРОВАНИЕ

Проектирование является неотъемлемой частью разработки любой программной системы. Условно этот процесс можно разделить на несколько последовательных, связанных друг с другом этапов: формулирование целей (сбор требований), анализ предметной области и создание прототипа системы. Анализ предметной области подразумевает более глубокое изучение требований, предъявляемых к функционалу проектируемой системы, проработку сценариев взаимодействия с системой и алгоритмов в бизнес-логике.

Четко сформулированные цели и задачи, тщательно проведенный анализ требований, проработка сценариев взаимодействия с системой и проектирование бизнес-логики – все это ключ успеха в достижении необходимого результата. Таким образом, проектирование увеличивает шансы успешного составления проекта, а также экономит время и деньги, ведь внесение изменений на начальном этапе разработки не так критично, как, например, на этапе отладки уже разработанной системы.

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		7

2 АНАЛИЗ ТРЕБОВАНИЙ

Задачей дипломного проекта было определено создание телеграмбота для построения маршрутов с использованием общественного транспорта города Киева.

Перед началом проектирования и разработки телеграмма бота нужно определить основные требования и описать их. Кроме того, необходимо проанализировать предметную область и рассмотреть существующие решения, чтобы убедиться в целесообразности разработки. Поскольку разрабатываемая система планируется внедрять в уже существующий программный комплекс, это накладывает определенные ограничения на архитектуру и технологии, которые можно будет использовать.

Исходя из поставленных целей, требуется разработка телеграмм бота, который: а) будет знать обо всех возможных остановках общественного транспорта и типе останавливаемого на них общественного транспорта;

б) будет хранить полученные данные в специальной базе данных, оптимизированной для хранения метрик и временных данных;

в) будет определенным образом сохранять маршруты и график работы общественного транспорта;

г) позволит отслеживать текущее местонахождение пользователя и общественного транспорта;

д) позволит получать информацию о прибытии общественного транспорта на остановку;

е) позволит строить оптимальные маршруты движения с использованием общественного транспорта;

ж) позволит находить ближайшие к местонахождению пользователя остановки общественного транспорта.

А теперь подробно опишем каждый из пунктов. Пользователю нужно будет знать все возможные остановки общественного транспорта в соответствии с указанным адресом.

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		8

Поскольку остановки общественного транспорта меняются не часто, мы можем хранить информацию о них в базе данных, чтобы не тратить время и ресурсы для поиска. Аналогично мы можем сделать с информацией о маршрутах и графиках общественного транспорта.

Также пользователь захочет видеть график прибытия общественного транспорта на остановку, тип и маршрут. Для того чтобы находить остановки общественного транспорта, находящиеся неподалеку от пользователя, мы должны получить доступ от пользователя на отслеживание его местонахождение.

Пользователю предоставляется возможность ввести нужный ему адрес или выбрать необходимую остановку и в зависимости от выбранного адреса или остановки телеграмм-бот строит для пользователя оптимальный маршрут передвижения по городу Киеву с использованием общественного транспорта.

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		9

3 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Проанализировав все возможные варианты создания данного чат-бота было обнаружено, что для написания такого бота необходимо использовать один из языков серверного программирования: Ruby, Node.JS, Python, PHP. Необходимо определить, какой именно язык лучше всего подходит для написания такого бота. Также важно умение работать с REST (Representational State Transfer) API (Application Programming Interface), которые предоставляют мессенджеры, а в данном случае это – TelegramBotAPI. После этого важно определиться с типом бота:

а) боты, имеющие способность к обучению (т.е. понимающие естественные языки), умеющие использовать логику при общении с пользователем и обрабатывать естественный язык для создания ответа на сообщения;

б) заскриптованные боты (боты, которые не способны понять естественный язык), у них диалог с пользователем – это заранее сформированный набор слов или предложений, а «скрипт» – это своеобразное дерево решений, в котором сценарий, запрограммированный заранее, является ответом на вопросы пользователя. Диалоги у них в большинстве случаев линейны и структурированы.

И в заключение, нужно определить цель бота, поскольку у него должна быть определенная цель, потому что в противном случае он не будет иметь смысла.

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		10

4 ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ

4.1. Приложение Eway

В ходе детального изучения предметной области, было найдено приложение "Eway" и соответствующий веб-сайт.

Поскольку приложения начинают терять популярность, большая часть людей начинает использовать телеграммы-боты в повседневной жизни, я решил выбрать тему: телеграмм-боты для построения маршрута с использованием общественного транспорта в городе Киеве. Поэтому для того чтобы создать телеграмм-бота, который будет иметь возможности web-сайта и приложения, нужно определить основные возможности приложения и web-сайта.

Приложение "Eway" подойдет, прежде всего, для того, чтобы отслеживать нужный вам транспорт на маршруте. Вы сможете выйти из дома и дойти до остановки как раз до прибытия. Единственный минус – пока еще не все транспортные средства снабжены GPS-маячками, поэтому их движение нельзя увидеть на карте. Но, что касается Киева, почти все маршруты «Киевпасстранса» и крупных транспортных компаний уже синхронизированы с приложением, правда, часть из них не отображается в режиме реального времени.

Пользоваться приложением несложно: нужно включить GPS на своем устройстве, и дальше Eway сам установит ваш город по умолчанию и найдет ближайшие остановки. Среди них выбирается нужная, далее – соответствующий маршрут из списка транспорта. Приложение показывает ожидаемое время прибытия транспорта конкретного маршрута на остановку на закладке с общей информацией, а также информацию об интервале движения транспорта, времени его работы и стоимости проезда. В подменю «Карта» видно, где именно находится нужный вам троллейбус/трамвай/маршрутка.

Среди других полезных функций следует отметить прокладку маршрутов с прорисовкой на карте города (метрополитен тоже учитывается). Кстати, сейчас сервис работает в меньшем количестве городов Украины, чем вначале, из-за конфликтов с компаниями, установившими права на

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		11

передачу GPS-данных. Отслеживание маршрутов есть в Киеве, Днепре, Каменском, Полтаве, Дрогобыче, Хмельницком, Кропивницком, Львове.

Среди преимуществ данного приложения можно выделить следующие: пользователь может получить необходимую информацию об общественном транспорте, его местонахождении, времени прибытия, ближайших остановках и возможности построить маршрут.

Основными недостатками являются:

- а) отсутствие GPS-маячков на всех транспортных средствах; б) медленная скорость работы;
- в) отсутствие телеграмм-бота.

Поскольку нет телеграмм-бота, который был бы связан с нашей предметной областью, мы рассмотрим телеграмм бот «Синоптик»@synbot.

4.2. Synbot

Synbot – это бот, главная цель которого состоит в информировании пользователя о последнем прогнозе погоды. Этот бот может предоставлять информацию о прогнозе погоды не только на текущий день, но и на несколько последующих.

Каждое утро люди интересуются прогнозом погоды. Иногда для получения последнего прогноза погоды с помощью обозревателя или приложения не хватает времени. Synbot минимизирует затраты времени на получение прогноза погоды, достаточно открыть приложение Telegram и войти в чат с ботом, который поможет помочь определиться с гардеробом. Именно благодаря быстрдействию этот бот становится все популярнее.

Рассмотрим основные функции данного бота. Как только мы открываем диалог с ботом, пользователю сразу предоставляется возможность узнать прогноз погоды на текущий день. Далее бот позволяет выбрать параметры отображения прогноза погоды, такие как: период времени, город, максимальная и

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		12

минимальная температура воздуха, преимущественное направление ветра и возможность осадков.

Если есть необходимость, пользователь может сохранить ранее введенные параметры и настроить автоматические оповещения, то есть сообщения от бота с прогнозом погоды.

Преимущества Synbot:

- а) удобство интерфейса;
- б) возможность регулировать и сохранять параметры; в) быстродействие;
- г) простой алгоритм действия; д) автосообщение.

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		13

Таблица 5.1.2. – Доступные команды для дополнительной настройки чат-бота

Команда	Описание
/ token	Возвращает полученный ранее токен у выбранного бота
/ revoke	Аннулирует токен доступа к боту
/ setinline	Включает или отключает возможность вызвать бота из других чатов
/setinlinegeo	Включает или выключает возможность передачи местоположения бота с другого чата
/setinlinefeedback	Позволяет получать информацию о количестве избранных пользователями команд
/setjoininggroup	Определяет, может ли быть добавлен в групповые диалоги бот
/setprivacy	Включает режим конфиденциальности. В этом режиме бот получает, обрабатывает и отправляет обратно информацию отдельно для каждого пользователя в чате.

После настроек на стороне Telegram и получения токена можно приступить к разработке программной части чат-бота.

5.2 Размещение чат-бота на удаленном сервере Heroku

Для постоянного и быстрого получения доступа к нашему чат-боту после разработки его необходимо разместить на локальном сервере. В качестве платформы был выбран Heroku. Размещение на облачной PaaS-платформе Heroku схоже с работой распределенной системы контроля версий (Git).

Изм.	Письмо	№ докум.	Подпись	Дата

Существует три способа развертывания:

- а) Heroku Git и Container Registry используя Heroku CLI; б) подключение аккаунта GitHub с автоматическим развертыванием; в) подключение аккаунта Dropbox с автоматическим развертыванием.

В работе был выбран способ и использование Heroku CLI.

а) для развертывания чат-боты были выполнены следующие

действия: б) регистрация на облачную SaaS-платформу Heroku;

в) загружен и установлен Heroku CLI;

г) в терминале операционной системы или встроенном в IDE выполнен вход в аккаунт Heroku командой «heroku login»;

д) клонированный репозиторий из удаленного Git-сервера Heroku на локальную машину с помощью команды `heroku git: clone -a <APP_NAME>`;

е) зафиксированные изменения в коде с помощью команд `git add`. И «`git commit -am <COMMIT_NAME>`»;

ж) все зафиксированные изменения были отправлены на удаленный сервер облачных вычислений Heroku командой `git push heroku master`.

После выполнения всех указанных команд в правильной последовательности начинается передача данных на удаленный сервер. Если развертывание прошло успешно, то на терминале отобразится справочная информация о состоянии программы и режиме доступа к нему.

Таким образом, чат-бот был развернут на удаленном сервере облачных вычислений Heroku, что позволит пользователям получать бесперебойный доступ.

5.3 Среда разработки

Visual Studio Code Visual Studio Code – это понятный, лаконичный, но многофункциональный редактор исходного кода, совместимый с Windows, MacOS и Linux.

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		16

IDE поддерживает большое количество популярных языков программирования.

Поддерживаемые расширения и среды выполнения представлены ниже:

- а) JavaScript;
- б) TypeScript;
- в) Node.js;
- г) C++;
- д) C#;
- е) Java;
- ж) Python;
- з) PHP;
- и) Go;
- к) .NET;
- л) Unity.

Visual Studio Code включает в себя встроенный отладчик, удобство для разработки программных продуктов, инструменты для работы с Git-репозиториями, подсветку синтаксиса, средства для рефакторинга и IntelliSense (технология автодополнения, которая предложит команду по первым буквам). Visual Studio Code был основан на фреймворке Electron, позволяющем используя Node.js разрабатывать настольные приложения, работающие на двигателе Blink.

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		17

5.4 Язык программирования – Python

Python – это язык программирования общего назначения, который направлен прежде всего на увеличение производительности самого разработчика ПО, чем кода, который он пишет.

Если говорить простым языком, Python позволяет написать практически все: веб-/настольные приложения, игры, автоматизированные информационные системы, комплексные системы, системы управления жизнеобеспечением и многое другое. Кроме того, порог вхождения низкий, а код во многом немногословен и понятен даже для того, кто никогда на нем не писал. За счет простоты кода, будущее сопровождение программ, написанных на Python, становится легче и приятнее по отношению к Java или C. А с точки зрения бизнеса это дает сокращение затрат и повышение эффективности трудовых ресурсов.

Практически на всех платформах и операционных системах был реализован интерпретатор Python, являющийся несомненным преимуществом. Первым таким языком был C, однако его типы данных на разных машинах довольно часто занимали разное количество памяти, а это представляет некое препятствие при написании больших программ.

Также важная черта – расширяемость языка, этому придается большое значение и, как пишет сам автор Гвидо ван Россум, язык был задуман именно как расширяемый. Это означает, что есть возможность усовершенствования языка всеми заинтересованными разработчиками. Интерпретатор был разработан на языке программирования C и исходный код доступен для любых изменений.

При необходимости можно вставить его в свою программу и использовать как собственную встроенную оболочку. Или же, написав на C дополнение к Python и скомпилировав приложение, получить "расширенный" интерпретатор с новыми возможностями.

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		18

Следующее значительное преимущество – наличие большого числа подключаемых модулей, обеспечивающих различные дополнительные возможности. Такие модули обычно пишутся на C, на самом Python, и могут быть созданы более опытными программистами. В качестве примера можно привести следующие модули:

а) Numerical Python – предоставляет программисту более расширенные математические возможности, такие как манипуляции с целыми векторами и матрицами;

б) Tkinter – позволяет строить приложения с использованием графического пользовательского интерфейса (GUI);

в) OpenGL – это большая библиотека графического моделирования двух- и трехмерных объектов Open Graphics Library.

Единственным недостатком, на который обращал внимание сам автор, является сравнительно невысокая скорость выполнения Python программы. Однако, это не играет большую роль по сравнению с преимуществами языка при написании программ не очень критичных к скорости выполнения. Из использованных модулей можно выделить: TeleBot, CherryPy, Requests, которые будут более подробно описаны ниже.

5.4.1 Модуль TeleBot

Модуль TeleBot является оболочкой над запросами к TelegramBotAPI, используется для упрощения и минимизации написанного кода. Все типы указаны в types.py. Все они полностью соответствуют определению типов API Telegram, за исключением from поля Message, которое переименовано в from_user (поскольку from это зарезервированный токен Python). К таким атрибутам, как message_id, можно обращаться непосредственно, например: message.message_id. При написании программ не очень требовательных к скорости выполнения стоит обратить внимание, что атрибут message.chat может принадлежать как определенному пользователю, так и групповому чату, это с полностью окупаемыми преимуществами языка.

						ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата			19

В классе TeleBot расположены все способы API. Чтобы следовать общим соглашениям об именах Python, они переименованы. Например: sendMessage, send_message, editMessageText, edit_message_text. Функция украшена декоратором экземпляра TeleBot – обработчик сообщений. Обработчики сообщений состоят из одного или нескольких фильтров. Каждый фильтр возвращает True или False для определенного сообщения и обработчик получает разрешение на обработку сообщения, если возвращается True.

5.4.2 Модуль CherryPy

Написанный на языке программирования Python, объектно-ориентированный веб-фреймворк – CherryPy. Создано для быстрой разработки веб-приложений для сети Интернет. Представляет собой надстройку над HTTP-протоколом.

Модуль может выступать в качестве самостоятельного веб-сервера или работать под управлением другой серверной программы на основе протокола WSGI. CherryPy не предназначен для обработки шаблонов для вывода данных, получения доступа к базе данных, авторизации пользователя. Фреймворк становится расширенным за счет фильтров, представляющих собой простые интерфейсы, вызываемые в определенных точках процесса обработки запросов/ответов и состоят из семи функций. Главным компонентом TurboGears является модуль CherryPy.

Процесс установки CherryPy совсем несложен, достаточно просто скачать его и выполнить несколько коротких шагов из файла README. Модуль CherryPy устанавливается так же, как любой другой модуль Python, поэтому не нужно указывать каталог установки. Если сравнивать с другими, более сложными серверными технологиями, модуль CherryPy всегда доступен для использования, как и любой другой модуль Python, если он будет импортирован с помощью инструкции import. В реальности CherryPy представляет собой не более чем самостоятельное приложение на языке Python, умеющее управлять своим собственным многопоточным веб-сервером, благодаря этому мы

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		20

имеем возможность выполнить сценарий на стороне сервера достаточно просто, как запустить простую команду в окне терминала. По умолчанию сервер запускается локально на 8080 порта, и сразу выполняет функцию, которую ему передали, но также можно задать определенные настройки вручную, такие как: IPадрес, порт, открытый и закрытый ключи SSL.

5.4.3 Модуль Requests

Requests – библиотека Python, выполняющая HTTP-запросы (HyperText Transfer Protocol). Начиная от передачи параметров в URL-адресах до отправки пользовательских заголовков и проверки SSL.

5.5 REST API, HTTP-запросы, JSON и череда сообщений

5.5.1 REST API

REST – это стиль архитектуры программного обеспечения, который используется для построения распределенных масштабируемых веб-сервисов, использующих запросы HTTP.

REST (Representational State Transfer) – стиль взаимодействия компонентов распределенного приложения в сети. REST имеет согласованный набор учитываемых ограничений при проектировании распределенной гипермедиа-системы. В определенных случаях (интернет-магазины, поисковые системы) это влечет за собой повышение производительности и изменение архитектуры, а именно ее упрощение. Компоненты в REST своим взаимодействием напоминают взаимодействие клиента и сервера в Интернете.

Вызов удаленной процедуры в сети Интернет может представлять собой обычный HTTP-запрос (обычно "GET" или "POST"; такой запрос называют "REST-запрос"), а все необходимые для передачи данные передаются в качестве параметров запроса.

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		21

Для веб-служб, построенных с учетом REST, применяют так называемый термин «RESTful».

В отличие от веб-сервисов на основе SOAP, не существует единого официального стандарта для термина RESTful веб-API. Поскольку REST является архитектурным стилем, в то время как SOAP является протоколом. REST не является стандартом сам по себе и, несмотря на это, большинство RESTful-реализаций используют известные стандарты, такие как: HTTP, URL, JSON и XML.

5.5.2 HTTP-запросы

Протокол Передачи Гипертекста (HTTP) – это один из известных протоколов стека TCP/IP (Transmission Control Protocol/Internet Protocol), с самого начала он был разработан для размещения и получения HTML (HyperText Markup Language) страниц и на данный момент используется для распределенных информационных систем. HTTP широко используется в Интернете, а именно для передачи информации и представляет собой наиболее используемый прикладной протокол.

HTTP это протокол, который определяется типом запрос/ответ. Когда клиент или веб-браузер отправляет сообщения с запросом на сервер, HTTP протокол самостоятельно определяет типы отправленных серверу сообщений, используемых клиентом для запроса на веб-страницу, а также типы сообщений, используемых сервером для ответа. Три наиболее популярных типа сообщений это: GET, POST и PUT. На рисунке 5.5.2.1 представлен пример GET-запроса.

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		22

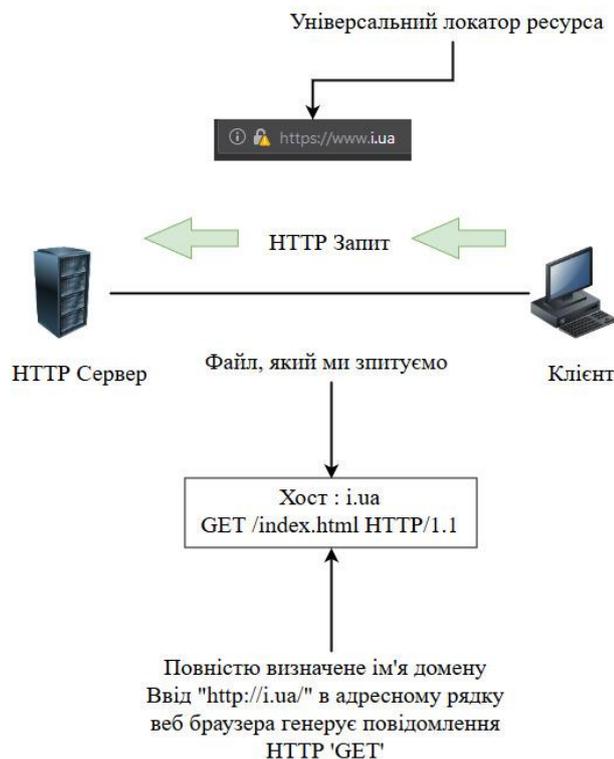


Рисунок 5.5.2.1 – HTTP протокол с использованием GET

Для того, чтобы отправить сообщения, загружающие данные на веб-сервер используются POST и PUT. Например, когда пользователь вводит данные в форму, находящуюся на веб-странице, POST добавляет эти данные в сообщения и они ссылаются на сервер. PUT загружает эти ресурсы или другой контент на веб-сервер.

HTTP протокол является очень гибким, но это не делает его безопасным протоколом. Сообщения POST отправляют данные на сервер в виде обычного текста, который может быть с легкостью перехвачен. Аналогично, ответы, поступающие от сервера, также не зашифрованы.

Для того чтобы создать безопасную коммуникацию через Интернет используется безопасный HTTP протокол (HTTPS), он позволяет получать доступ или публиковать информацию на веб сервере. HTTPS позволяет использовать аутентификацию и шифрование, чтобы защитить данные от перехвата при перемещении между клиентом и сервером. HTTPS

определяет некоторые правила для прохождения данных между прикладным и транспортным уровнями.

5.5.3 Формат JSON

JSON (JavaScript Object Notation) – это формат для обмена данными, комфортный для хранения, чтения и написания как человеком, так и компьютером. Он был основан на подмножестве известного языка программирования JavaScript и определен в стандарте ECMA-262 3rd Edition - December 1999. JSON – представляет собой почти обычный текстовый формат, полностью независимый от языка реализации, но он использует некоторые правила, знакомые программистам C-подобных языков, таких как C, C++, C#, Java, JavaScript, Perl, Python и многих других. Именно эти свойства делают JSON идеальным языком, который можно использовать для обмена данными.

JSON с самого начала был основан на двух структурах данных: а) коллекция пар ключ/значение. Эта концепция в разных языках может быть реализована как объект, запись, структура, словарь, хэш, именуемый список или ассоциативный массив;

б) упорядоченный перечень значений. Большинство языков программирования реализуют это в качестве массива, вектора, списка или последовательности. Это универсальные структуры данных. Почти все современные языки программирования могут поддерживать их в любой форме. Достаточно легко предположить, что формат данных, хранящихся в JSON, независим от языка программирования и должен быть основан на структурах, указанных выше.

В нотации JSON это выглядит так:

а) объект – это неупорядоченный набор пар ключ/значения. Объект начинается с открытой фигурной скобки и оканчивается закрытой фигурной скобкой. Каждое имя в объекте сопровождается двоеточием, а пара ключ/значения разделяются запятой (рисунок 5.5.3.1);

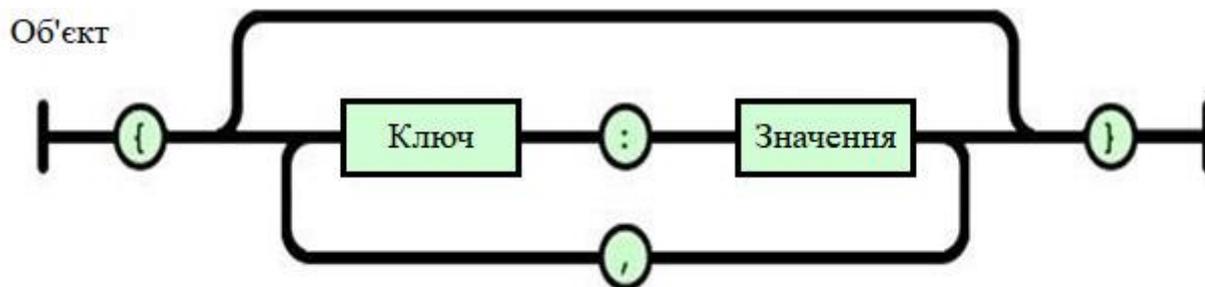


Рисунок 5.5.3.1 – Схематическое представление объекта JSON

б) массив – это упорядоченная коллекция, которая может содержать разные значения. Массив начинается с квадратной скобки и заканчивается этой скобкой. Значения в массиве разделяются запятой (рисунок 5.5.3.2);



Рисунок 5.5.3.2 – Схематическое представление массива JSON

в) значение – может быть числом, строкой в двойных кавычках, true, false, null, объектом или массивом. Эти структуры могут быть вложены (рисунок 5.5.3.3);

г) строка – это так называемый набор символов в кодировке Unicode, заключенный в двойные кавычки, или символ, использующий обратный слэш в качестве символа экранирования. Символ представляется как одна символьная строка. Сходный синтаксис используется в языках программирования C и Java (рисунок 5.5.3.4);

Число в JSON передается так, как в C или Java, кроме того, что используется только десятичная система счисления (рисунок 5.5.3.5).

Изм.	Письмо	№ докум.	Подпись	Дата
------	--------	----------	---------	------

Значення

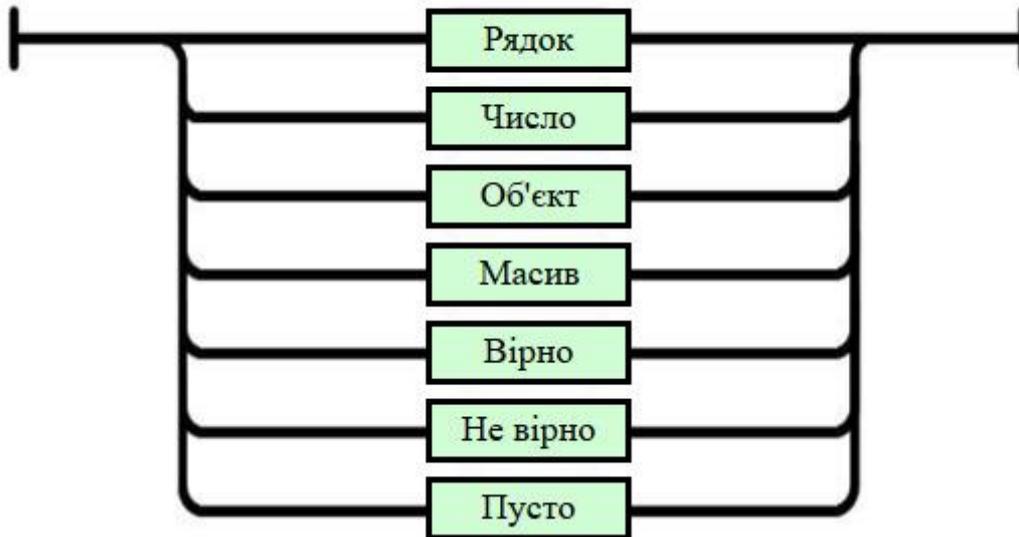


Рисунок 5.5.3.3 – Схематическое представление значения в JSON



Рисунок 5.5.3.4 – Схематическое представление строки в JS

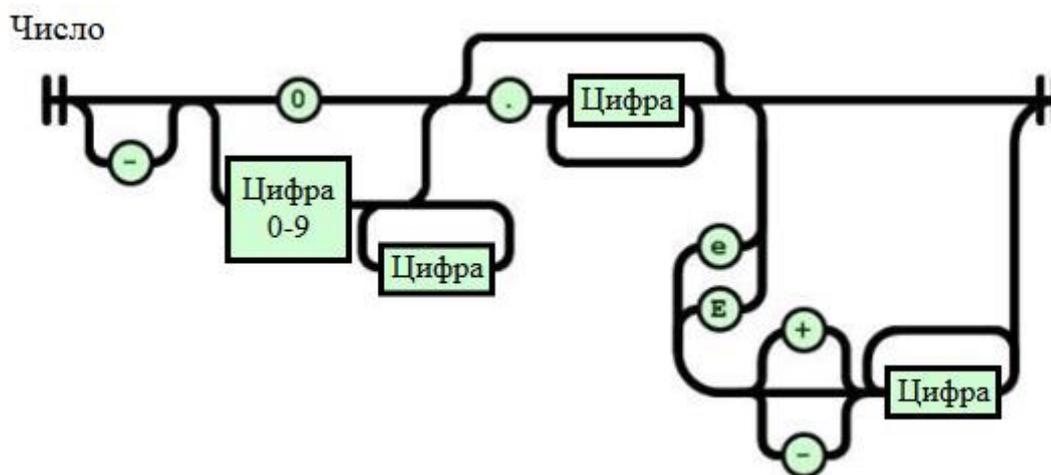


Рисунок 5.5.3.5 – Схематическое представление числа в JSON

5.5.4 Платформа Heroku

Heroku представляет собой облачную PaaS-платформу, которая поддерживает большое количество языков программирования. С 2010 г. она становится дочерней компанией Salesforce. Heroku, это одна из первых облачных платформ, появившаяся в июне 2007 года и в начальной версии поддерживавшей только один язык программирования Ruby, но на данный момент список поддерживаемых языков расширился и включает в себя Java, Node.js, Scala, Clojure, Python, Go и PHP. Обычно на серверах используются операционные системы Debian или Ubuntu. В 2011 году была введена поддержка Node.js и Clojure. Также платформа может поддерживать такие СУБД, как Cloudant, Membase, MongoDB и Redis, кроме PostgreSQL.

Приложения, работающие на данной облачной платформе, используют также DNS-сервер (обычно приложения имеют личное доменное имя, которое выглядит так – название проекта.herokuapp.com).

Для всех программ выделяются несколько независимых виртуальных процессов, которые называются dynos. Они размещены по специальной виртуальной сетке, состоящей из нескольких серверов. Одной из особенностей Heroku является то, что у нее есть собственная система контроля версий.

5.5.5 Очередь сообщений

Очередь сообщений (message queue, MQ) – это архитектура обмена сообщениями между разными компонентами в программных системах в асинхронном режиме. Это позволяет отправлять сообщения одним компонентом-отправителем системы в одно время, а получить и обработать другим компонентом-получателем в совсем другой. Системы такого типа состоят из producer'a (отправителя) и consumer'a (получателя), взаимодействующих между собой с помощью поставщика (broker), который по-другому можно определить как некоторый MQ-сервер.

Очередь – это структура данных с ограниченным доступом к элементам, которую можно описать, как «первый пришел-первый вышел». Ограниченность состоит в том, что элемент можно добавить только в конец очереди, а выбрать элемент только из начала, при удалении элемента из очереди он уничтожается.

Особенности использования MQ-архитектуры:

а) слабое связывание – создаются неявные интерфейсы, которые обмениваются данными, позволяют процессам быть независимыми друг от друга и позволяют сохранять желаемый формат сообщений;

б) избыточность – позволяет избегать нерационального использования ресурсов системы или сети, в свою очередь сохраняя еще необработанную информацию;

в) масштабируемость – посредством распределения процессов обработки информации позволяет увеличивать производительность MQ-сервера;

г) эластичность и возможность выдерживать пиковые нагрузки – при высокой нагрузке в дополнение, очереди сообщений могут служить так называемым буфером для накопления информации, позволяя изменить скорость обработки информации и, тем самым, снизить общую нагрузку на систему или сеть;

д) устойчивость – дает возможность отделить один процесс от друга и получение сообщений, благодаря этому, в случае отказа процесса-обработчика, предоставляется возможность восстановить рабочее состояние системы, откладывая обработку сообщения;

е) гарантированная доставка – сообщения в любом случае будут обработаны и доставлены, независимо от рабочего состояния системы отправителя и системы получателя сообщений. Это достигается использованием асинхронной связи и возможностью хранения сообщения на MQ-сервере, пока оно не будет обработано;

ж) гарантированный и одноразовый порядок доставки – значительная часть MQ-серверов позволяет доставить данные именно в том порядке, в котором они были отправлены, при этом гарантирует, что после того, как сообщение будет прочитано, оно будет удалено из очереди;

з) буферизация – это гарантия доставки сообщений с максимальной эффективностью, которая достигается посредством независимости скорости обработки сообщений от скорости их отправки, это в свою очередь происходит благодаря использованию вышеупомянутой структуры очереди – своеобразного буфера между системой-обработчиком и системой. отправителем.

5.6 Telegram Bot API

Bot API представляет собой HTTP-интерфейс для работы с ботами в Telegram. Каждый бот – это специально созданный аккаунт для автоматической обработки и отправки сообщений.

Документация Telegram Bot API выделяет два максимально противоположных способа получения обновлений:

а) периодические запросы;

б) установка веб-хуков.

Входящие обновления сохраняются до тех пор, пока сервер не обработает его, но не более 24 часов. В ответ получаем объект

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		29

Update, который сериализован в JSON, независимо от способа получения обновлений.

Первый и наиболее простой вариант заключается в периодическом опросе серверов Telegram на наличие новой информации. Открывается соединение на непродолжительное время и все обновления сразу отправляются боту, все это осуществляется через связь Long Polling. Этот способ простой, но не очень надежный.

Веб-хуки работают несколько иначе. Если в чат приходит сообщение, то Telegram сам говорит об этом, в этом и заключается работа вебхука. Теперь отпадает необходимость периодически опрашивать серверы, тем самым исчезает причина ошибок поисковых роботов. Однако за эту возможность нужно платить необходимостью установки полноценного веб-сервера на то устройство, на котором и планируется запускать поисковых ботов.

Так же для корректной работы необходимо иметь собственный сертификат SSL (Secure Sockets Layer), поскольку веб-хуки в Telegram работают только по HTTPS (рисунок 5.6.1).

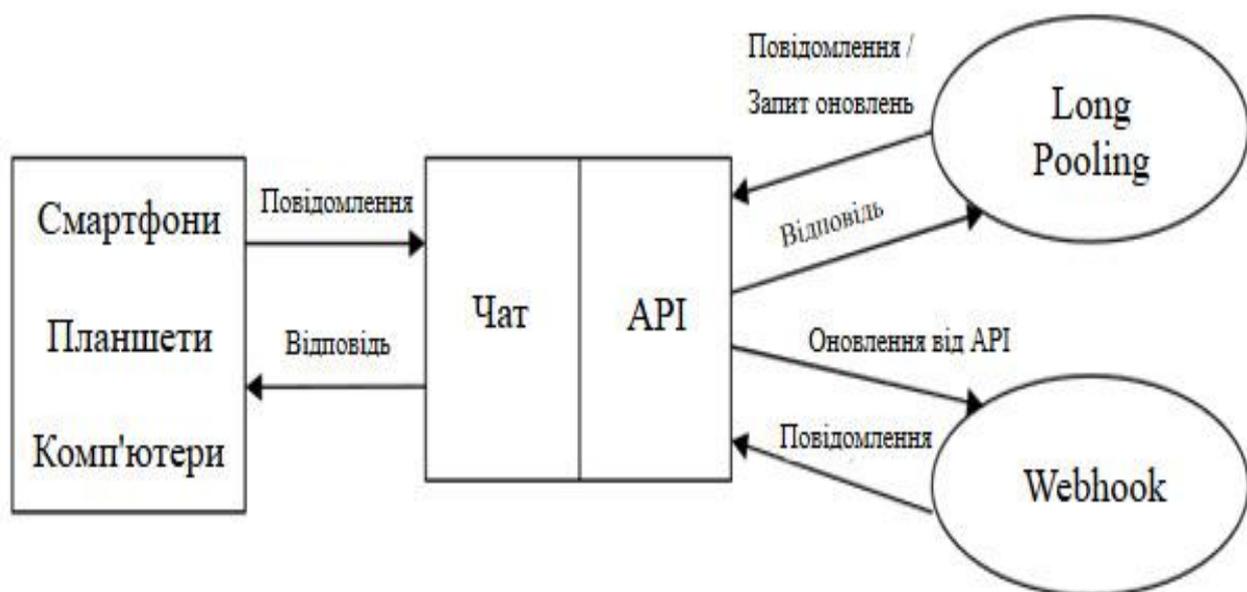


Рисунок 5.6.1 – Принцип работы чат-бота на платформе Telegram

Для получения токена необходимо написать специальному боту @BotFather.

Ниже описаны примеры доступных методов для API:

а) `getUpdates` – это метод, используемый для получения обновлений по технологии long polling;

б) `setWebhook` – этот способ привязывает к роботу url домена, где содержится текущий запущенный бот;

в) `sendMessage` – метод, отправляющий текстовое сообщение в клиентской части Telegram;

г) `sendLocation` – метод, отправляющий сообщения с координатами местоположения устройства;

д) `getFile` – метод, возвращающий вложенные файлы по их имени.

Допускаются POST и GET запросы.

Есть 4 метода передачи характеристик в Bot API: а)

запрос в URL;

б) `application/x-www-form-urlencoded`; в)

`application/json`;

г) `multipart/form-data` (подходит для загрузки файлов).

Для работы с Telegram Bot API была исследована документация, в которой описаны все способы и характеристики, было установлено, что все ответы приходят в JSON-формате. В ходе написания чат-бота были протестированы и использованы следующие методы и типы:

Метод `getUpdates` используется для получения обновлений через long polling.

Ответ возвращается посредством массива объектов Update. Параметры метода приведены в таблице 5.6.1.

										Письмо
Изм.	Письмо	№ докум.	Подпись	Дата						31

Таблица 5.6.1 – Параметры getUpdates

Параметры	Тип	обязательный	Описание
Offset	Integer	Нет	Смещение относительно начала массива Update, разрешающее просматривать новые сообщения, игнорируя старые.
Limit	Integer	Нет	Ограничивает количество элементов массива Update. Принимает значение от 1 до 100.
Timeout	Integer	Нет	Таймаут в секундах для продолжительного опроса. По умолчанию 0, то есть так называемый краткий запрос.

Метод setWebhook необходим для задания URL веб-хука, на который бот будет отправлять обновления. Каждый раз при получении обновления по этому адресу будет отправлен HTTPS POST с сериализованным в JSON объектом Update.

При ошибке запроса на сервер попытка будет повторена несколько раз. Для большей безопасности рекомендуется включить токен в URL веб-хука, например: [https://yourwebhookserver.com/ <token>](https://yourwebhookserver.com/<token>). Мы можем быть уверены, что запросы к веб-хуку отправляет именно Telegram, поскольку никто посторонний не знает токена. Параметры данного метода приведены в таблице 5.6.2

Таблица 5.6.2 – Параметры setWebhook

Параметры	Тип	обязательный	Описание
url	String	Нет	HTTPS url для отправки запросов.
Sertificate	InputFile	Нет	Загрузка общественного ключа для проверки корневого сертификата.

Метод `getWebhookInfo` включает в себя информацию о текущем состоянии веб-хука. Параметры данного метода приведены в таблице 5.6.3.

Таблица 5.6.3 – Настройки возвращаемые `getWebhookIn`

Параметры	Тип	Описание
<code>url</code>	String	url веб-хука, может быть пустым.
<code>has_custom_certificate</code>	Boolean	Да, только если веб-хук использует свой самоверенный сертификат.
<code>pending_update_count</code>	Integer	Количество обновлений, ожидающих своей очереди.
<code>last_error_date</code>	Integer	Не обязательно. Unix – время последней ошибки доставки обновления на указанный веб-хук.
<code>last_error_message</code>	String	Не обязательно. Описание последней ошибки доставки обновления на указанный веб-хук.

Метод `sendMessage` используется для отправки сообщений. Параметры метода приведены в таблице 5.6.5.

Таблица 5.6.4 – Параметры `sendMessage`

Параметры	Тип	Обязательный	Описание
<code>chat_id</code>	Integer или String	Да	Уникальный идентификатор целевого чата или имени целевого канала.
<code>Text</code>	String	Да	Текст сообщения, <small>какое</small> необходимо отправить.

Продолжение таблицы 5.6.4.

Параметры	Тип	Обязательный	Описание
parse_mode	String	Нет	Нужно отправить Markdown или HTML, чтобы приложения Telegram отражали полужирный, курсивный, текст с фиксированной шириной или встроенные URL-адреса в сообщении бота.
disable_web_page_preview	Boolean	Нет	Отключает предварительный просмотр ссылок в сообщении.
disable_notification	Boolean	Нет	Отправляет сообщение тихо. Пользователи iOS не получают сообщение, пользователи Android получают сообщение без звука
reply_to_message_id	Integer	Нет	Идентификатор выходного сообщения.
reply_markup	InlineKeyboard Markup или ReplyKeyboard Markup или ReplyKeyboard Hide	Нет	Расширенные возможности поиска интерфейса. Представляет собой JSON-сериализованный объект для встроенной клавиатуры, какая пользовательская.

Метод sendPhoto используется для отправки фотографий. Параметры метода приведены в таблице 5.6.5.

Таблица 5.6.5 – Настройки sendPhoto

Параметры	Тип	Обязательный	Описание
chat_id	Integer или String	Да	Уникальный идентификатор целевого чата или имя целевого канала.
Photo	InputFile или String	Да	Фото для отправки. Можно передать file_id как строку, чтобы отправить фотографию которая уже находится на серверах Telegram, или скачать новую фотографию.
Параметры	Тип	Обязательный	Описание
Caption	String	Нет	Тема фотографии от 0 до 200 символов.
replay_to message_id	Integer	Нет	Идентификатор исходящего сообщения.

Метод editMessageText используется для редактирования текстовых сообщений, отправленных ботом или через бота. Параметры метода приведены в таблице 5.6.6.

Продолжение таблицы 5.6.5.

Параметры	Тип	Обязательный	Описание
Replay_markup	InlineKeyboard Markup or ReplyKeyboard Markup or ReplyKeyboard Hide or ForceReply	Нет	Расширенные возможности поиска интерфейса. Представляет из себя JSON-сериализованный объект для встроенной клавиатуры, предназначенной для пользователя, определенные инструкции которые используются, для сокрытие клавиатуры пользователя ИЛИ принудительного ответа от пользователя.

Таблица 5.6.6 – Настройки метода editMessageText

Параметры	Тип	Обязательный	Описание
chat_id	Integer или String	Нет	Необходим параметр, если inline_message_id не указан. Уникальный идентификатор целевого чата или имени канала.
inline_message_id	String	Нет	Обязательно если chat_id и message_id не указаны. Идентификатор встроенного сообщения

Продолжение таблицы 5.6.6.

Параметры	Тип	Обязательный	Описание
Text	String	Да	Новый текст сообщения.
parse_mode	String	Нет	Нужно отправить Markdown или HTML, чтобы приложения Telegram отражали полужирный, курсивный, текст из фиксированной шириной ИЛИ встроенные URL-адреса В сообщении нашего сапога.
disable_web_page_preview	Boolean	Нет	Отключает предварительный просмотр ссылок для ссылок в этом сообщении.
replay_markup	InlineKeyboard Markup или ReplyKeyboard Markup или ReplyKeyboard Hide	Нет	Расширенные возможности поиск интерфейса.

Объект типа User предоставляет информацию о пользователе Telegram. Поля типа приведены в таблице 5.6.7

Таблица 5.6.7 – Поля объекта User

Поле	Тип	Описание
Id	Integer	Уникальный идентификатор бота или пользователя
first_name	String	Имя бота или пользователя
last_name	String	Не обязательно. Фамилия бота или пользователя
Username	String	Не обязательно. Username бота или пользователя

Объект типа Chat представляет собой информацию о чате. Поля типа приведены в таблице 5.6.8

Таблица 5.6.8 – Поля объекта Chat

Поле	Тип	Описание
Id	Integer	Уникальный идентификатор чата. Абсолютное значение не превышает 1e13
Type	Enum	Тип чата: а) private; б) group; в) supergroup; г) channel;
Title	String	Не обязательно. Название для каналов или групп
Username	String	Не обязательно. Username, для чатов и некоторых каналов
first_name	String	Не обязательно. Имя собеседника в чате
last_name	String	Нет Обязательно. Фамилия собеседника в чате
all_members_are_administrators	Boolean	Не обязательно. True, если все чат-участники являются администраторами.

Объект типа Message представляет собой информацию о сообщениях. Поля типа приведены в таблице 5.6.9.

Таблица 5.6.9 – Поля объекта Message

Поле	Тип	Описание
message_id	Integer	Уникальный идентификатор сообщения
From	User	Не обязательно. Отправитель. Может быть пустым в каналах.
Date	Integer	Дата отправки сообщения (Unix time)
Chat	Chat	Диалог, в котором было отправлено сообщение
forward_from	User	Не обязательно. Для пересланных сообщений: отправитель оригинального сообщения
forward_date	Integer	Не обязательно. Для отправки сообщений: дата отправки сообщения.
reply_to_message	Message	Не обязательно. Ответ: оригинальное сообщение. Следует обратить внимание, что объект Message в этом поле не будет содержать дополнительных полей reply_to_message, даже если он сам является ответом.
Text	String	Не обязательно. Для текстовых сообщений: текст сообщения, 0-4096 символов.

Продолжение таблицы 5.6.9.

Поле	Тип	Описание
Entities	Массив с MessageEntity	Не обязательно. Для текстовых сообщений: особое существо в тексте сообщения.
Поле	Тип	Описание
Document	Document	Не обязательно. Информация о фото
Photo	массив с PhotoSize	Не обязательно. Доступные размеры фото
Sticker	Sticker	Не обязательно. Информация о стикерах
Video	Video	Не обязательно. Информация о видеозаписи
Voice	Voice	Не обязательно. Информация о голосовом сообщении
Caption	String	Не обязательно. Подпись к файлу, фото или видео, 0-200 символов
Contact	Contact	Не обязательно. Информация об отправленном контакте.
Место	Место	Не обязательно. Информация о местонахождении.
Venue	Venue	Не обязательно. Информация о местоположении на карте.
new_chat_member	User	Не обязательно. Информация о пользователе, добавленном в группу.
left_chat_member	User	Не обязательно. Информация о пользователе, которого удалили.

Продолжение таблицы 5.6.9.

Поле	Тип	Описание
new_chat_title	String	Не обязательно. Название группы было изменено на поле.
new_chat_photo	массив с PhotoSize	Не обязательно. Фото группы было изменено на это поле.
Поле	Тип	Описание
group_chat_created	True	Нет Обязательно. Сервисное сообщение: группа создана
supergroup_chat_created	True	Нет Обязательно. Сервисное сообщение: супергруппа создана.
channel_chat_created	True	Нет Обязательно. Сервисное сообщение: канал создан.
migrate_to_chat_id	Integer	Не обязательно. группа была превращена в супергруппу с указанным идентификатором. Не превышает 1e13
migrate_from_chat_id	Integer	Не обязательно. Супергруппа была создана из группы с указанным идентификатором. Нет превышает 1e13.
pinned_message	Message	Нет Обязательно. Указанное сообщение было прикреплен. Следует обратить внимание, что объект Message в этом поле не будет содержать дополнительные поля reply_to_message, даже если он сам является ответом

Объект типа ReplyKeyboardMarkup представляет собой клавиатуру с опциями ответа. Поля типа приведены в таблице 5.6.10.

Таблица 5.6.10 – Поля объекта ReplyKeyboardMarkup

Поле	Тип	Описание
Keyboard	массив массивов С KeyboardButton	Массив рядов кнопок, каждый из которых является массивом объектов KeyboardButton
resize_keyboard	Boolean	Нет Обязательно. Указывает клиенту подогнать высоту клавиатуры под количество кнопок (сделать его меньше, если кнопок мало). По умолчанию False, то есть клавиатура всегда такого же размера, как стандартная клавиатура устройства.
one_time_keyboard	Boolean	Не обязательно. Указывает клиенту скрыть клавиатуру после использования (по нажатию на кнопку). Ее по-прежнему можно будет открыть через иконку в поле ввода сообщения. По умолчанию False.
Selective	Boolean	Не обязательно. Этот параметр требуется для показа клавиатуры только некоторым пользователям. Пример: пользователь отправляет запрос на изменение языка бота. Бот отправляет клавиатуру со списком языков, видимую только этому пользователю.

Объект типа KeyboardButton является еще одной кнопкой в ответной клавиатуре. Для обычных текстовых кнопок этот объект может изменяться на строку, содержащую текст на кнопке. Поля типа приведены в таблице 5.6.11.

Таблица 5.6.11 – Поля объекта KeyboardButton

Поле	Тип	Описание
Text	String	Текст на кнопке. Если ни одно С опциональные поля не использованы, то при нажатии на кнопку этот текст будет отправлен боту как простое сообщение.
request_contact	Boolean	Не обязательно. Если значение True, то при нажатии на кнопку бота отправится пользовательский контакт с его номером телефона. Доступно только в диалогах с ботом.
request_location	Boolean	Не обязательно. Если значение True, то при нажатии на кнопку бота отправится местоположение пользователя. Доступно только в диалогах с ботом.

Объект InlineKeyboardMarkup представляет собой встроенную клавиатуру, которая появляется под соответствующим сообщением. Поля типа приведены в таблице 5.6.12.

Объект типа InlineKeyboardButton – это еще одна кнопка встроенной клавиатуры. Мы обязательно должны задействовать ровно одно опциональное поле. Поля типа приведены в таблице 5.6.13.

Таблица 5.6.12 – Поля объекта InlineKeyboardMarkup

Поле	Тип	Описание
inline_keyboard	Массив массивов C InlineKeyboardButton	Массив строк, каждая из которых является массивом объектов InlineKeyboardButton.

Таблица 5.6.13 – Поля объекта Inline KeyboardButton

Поле	Тип	Описание
Text	String	Текст на кнопке
url	String	Не обязательно. URL, который откроется при нажатии на кнопку
callback_data	String	Не обязательно. Данные, которые будут отправлены в callback_query при нажатие на кнопку.
switch_inline_query	String	Не обязательно. Если этот параметр задан, то при нажатии кнопки дополнение предложит пользователю выбрать любой чат, откроет его и вставит в поле ввода сообщения юзернейм бота и определенный запрос для встроенного режима. Если отправлять пустое поле, то будет вставлен только юзернейм бота.

Продолжение таблицы 5.6.13

Поле	Тип	Описание
switch_inline_query_current_chat	String	<p>Нет Обязательно. Если установлено, нажатие кнопки введет имя бота и указан встроенный запрос в поле ввода текущего чата. Может быть пустым, и в этом случае будет вставлено только ИМЯ пользователя бота.</p>
callback_game	CallbackGame	<p>Описание игры, которая будет запущена, когда пользователь нажмет кнопку.</p> <p>Примечание: Этот тип кнопки всегда должен быть первой кнопкой в первой строке.</p>

6 БАЗА ДАННЫХ – MONGODB

MongoDB – это база данных, которая является документоориентированной с открытым исходным кодом и не требует описания схем таблиц. Главные черты и возможности:

- а) поддержка отказоустойчивости и масштабируемости;
- б) асинхронная репликация, набор реплик и распределения базы данных по узлам; в) JSON-подобная схема хранения данных;
- г) эффективное хранение больших объектов, административный интерфейс, серверные функции, Map/Reduce и прочее;
- д) использование Javascript в качестве языка для формирования запросов; е) профилирование запросов;
- ж) широкий набор атомарных операций над данными (условный поиск, сложная вставка/обновление и т.п.);
- з) разные типы данных (в том числе поддержка массивов); поддержка индексов (B-Tree);
- и) полнотекстовый поиск, в том числе на украинском языке, с поддержкой морфологии;
- к) журнал операций, изменяющих данные в базе данных.

MongoDB предлагает документоориентированную модель данных в отличие от реляционных баз данных, благодаря чему она работает в разы быстрее, имеет лучшую масштабируемость и ее легче использовать. Однако, учитывая все недостатки традиционных баз данных и преимущества MongoDB, важным пониманием является то, что задачи бывают разные и методы их решения бывают также разные.

В определенной ситуации MongoDB действительно способна улучшить производительность программы, например возьмем ситуацию, которая существует потребность хранить сложные по структуре данные. В другой ситуации лучше будет использовать традиционные реляционные базы данных.

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		46

MongoDB может представлять не только единственную базу данных, находящуюся на одном физическом сервере. Функциональность MongoDB позволяет разместить несколько баз данных на нескольких физических серверах, и эти базы смогут легко обмениваться данными и сохранять целостность.

В процессе создания телеграмма была спроектирована схема взаимодействия пользователя с базой данных, которая дает ответы на его сообщение (рисунок 6.1).

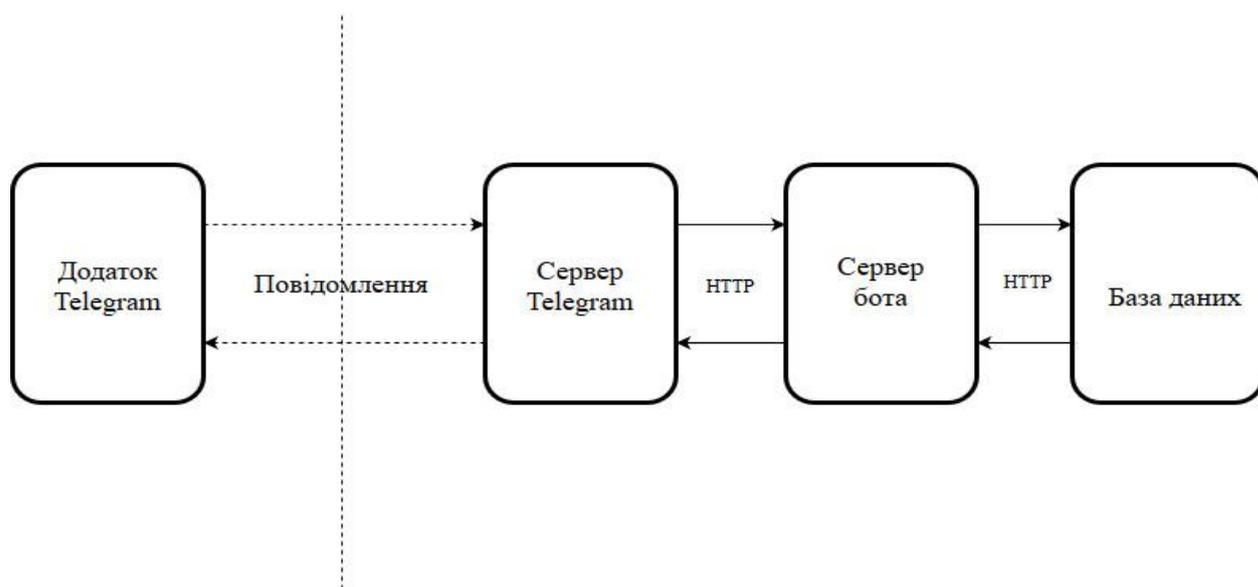


Рисунок 6.1 – Схема взаимодействия пользователя с базой данных

Вот несколько важных свойств этой базы данных:

а) формат данных в MongoDB. Одним из популярных стандартов обмена данными и их хранения является JSON (JavaScript Object Notation). JSON эффективно описывает сложные по структуре данные. Способ хранения данных в MongoDB в этом плане похож на JSON, хотя формально JSON не используется. Для хранения в MongoDB применяется формат, который называется BSON или сокращение от binary JSON. BSON позволяет работать с данными быстрее: быстрее производится поиск и обработка. Хотя следует отметить, что BSON в отличие от хранения данных в формате JSON имеет небольшой недостаток: в целом данные в JSON-формате занимают меньше места, чем в формате BSON, с другой стороны, данный недостаток с лихвой окупается скоростью;

б) кроссплатформенность. MongoDB написана на C++, поэтому ее легко перенести на разные платформы. MongoDB может быть развернута на платформах Windows, Linux, MacOS, Solaris. Можно скачать исходный код и самому скомпилировать MongoDB, но рекомендуется использовать библиотеки с официального сайта;

в) коллекции. Если в традиционном мире SQL есть таблицы, то в мире MongoDB есть коллекции. И если в реляционных БД таблицы хранят однотипные жестко структурированные объекты, то в коллекции могут содержать самые разные объекты, имеющие разную структуру и разный набор свойств;

г) репликация. Система хранения данных в MongoDB представляет набор репликов. В этом наборе имеется основной узел и может быть набор вторичных узлов. Все вторичные узлы сохраняют целостность и автоматически обновляются вместе с обновлением главного узла. И если основной узел по каким-то причинам выходит из строя, то один из вторичных узлов становится главным;

д) простота в использовании. Отсутствие жесткой схемы базы данных и в связи с этим потребности при малейшем изменении концепции хранения данных вновь создавать эту схему значительно облегчают работу с базами данных MongoDB и последующим их масштабированием. Кроме того, экономится время разработчиков. Им больше не нужно думать о новом создании базы данных и тратить время на построение сложных запросов;

е) GridFS. Одной из проблем при работе с любыми системами баз данных есть сохранение данных большого размера. Можно хранить данные в файлах, используя разные языки программирования. Некоторые СУБД предлагают специальные типы данных для хранения бинарных данных в БД (например, BLOB в MySQL). В отличие от реляционных СУБД MongoDB позволяет хранить разные документы с разным набором данных, однако при этом размер документа ограничивается 16 Мб. Но MongoDB предлагает решение – специальную технологию GridFS, которая позволяет хранить данные по размеру больше, чем 16 Мб. Система GridFS состоит из двух коллекций. В первой коллекции, которая

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		48

называется files, сохраняются имена файлов, а также их метаданные, например размер. А в другой коллекции, которая называется chunks, в виде небольших сегментов хранятся данные файлов обычно сегментами по 256 Кб.

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		49

7 TELEGRAM

Telegram – современный популярный сервис по обмену информацией. Кроме версии для браузеров, существуют также версии для ОС Linux, Mac и Windows, а также для популярных мобильных ОС, таких как Android, iOS и Windows Phone.

Одной из главных функциональных возможностей этого сервиса является возможность создания сторонних приложений, так называемых «ботов», которые могут взаимодействовать с пользователями, отвечать на любые пользовательские команды, а также отправлять им сообщения. К примеру, уже сегодня существует множество ботов для получения новостей, последнего прогноза погоды, для создания заметок и напоминаний, для проведения банковских операций и т.д.

Сообщения, команды и запросы, отправленные пользователями, передаются на сервер, а именно разработчикам поисковых роботов. Сервер Telegram является только посредником и предоставляет свой API для разработчиков, доступный с помощью HTTPS-интерфейса, предлагающего упрощенную версию API Telegram, так называемый Bot API. Для того чтобы начать взаимодействовать с ботом, пользователь должен сначала добавить его к себе в чат или в группу. или отправляя команды из чата непосредственно какому-либо боту, используя символ @ перед его именем в сообщении с командой. Детальнее о ботах можно почитать на официальном сайте Telegram, в данной работе нас интересует только возможность добавления бота в чат или группу, после чего он сможет посылать туда оповещения при любых отклонениях анализируемых метрик.

Следует уточнить этап создания нового бота и использования API, предоставляемого Telegram. Согласно инструкции с официального сайта, для создания бота минимально необходимы следующие пункты:

а) добавить бота «BotFather» к себе в чат, позволяющий управлять созданием и настройкой новых ботов внутри Telegram;

										Письмо
Изм.	Письмо	№ докум.	Подпись	Дата	ИТ51.320БАК.002 ПО					50

б) отправить боту команду «/newbot», после чего тот запросит полное и короткое имя создаваемого бота (полное имя используется для отображения пользователю названия бота в заголовке чата, короткое имя для ссылки быстрых команд) и сгенерирует уникальный токен авторизации, который потребуется для идентификации бота и взаимодействия через TelegramAPI при отправке запросов.

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		51

8 ТЕСТИРОВАНИЕ СИСТЕМЫ

Тестирование телеграмм бота производилось вручную. После каждого этапа разработки выполнялось тестирование работоспособности по заранее подготовленным тест-кейсам, которые были составлены при определении целей разработки и используемых инструментов.

Работа с Telegram ботом:

- а) скорость отклика бота на сообщение о начале работы;
- б) корректная обработка сообщений разного типа от пользователя.
- в) корректное отображение диалога;
- г) обработка нажатия на клавишу отправки сообщения; д) скорость отображения ответа после того, как пользователь написал сообщение;
- е) скорость работы базы данных; ж) оценка корректности ответов.

Тестирование бота производилось на мобильном устройстве Apple Iphone 7 Plus со следующими техническими характеристиками.

- а) экран: 5,5", IPS LCD, 1920x1080, мультитач; б) процессор: Apple A10 Fusion, 4x1;
- в) операционная система: iOS 12.1.4;
- г) оперативная память: 3 ГБ;
- д) встроенная память: 128 ГБ; е) навигация: GPS;
- ж) Telegram: Telegram v 5.7.

После подключения к чату пользователю доступна только одна команда – start. Отправив эту команду, в ответ получаем основную информацию о ботах (рисунок 8.1).

										Письмо
Изм.	Письмо	№ докум.	Подпись	Дата	ИТ51.320БАК.002 ПО					52

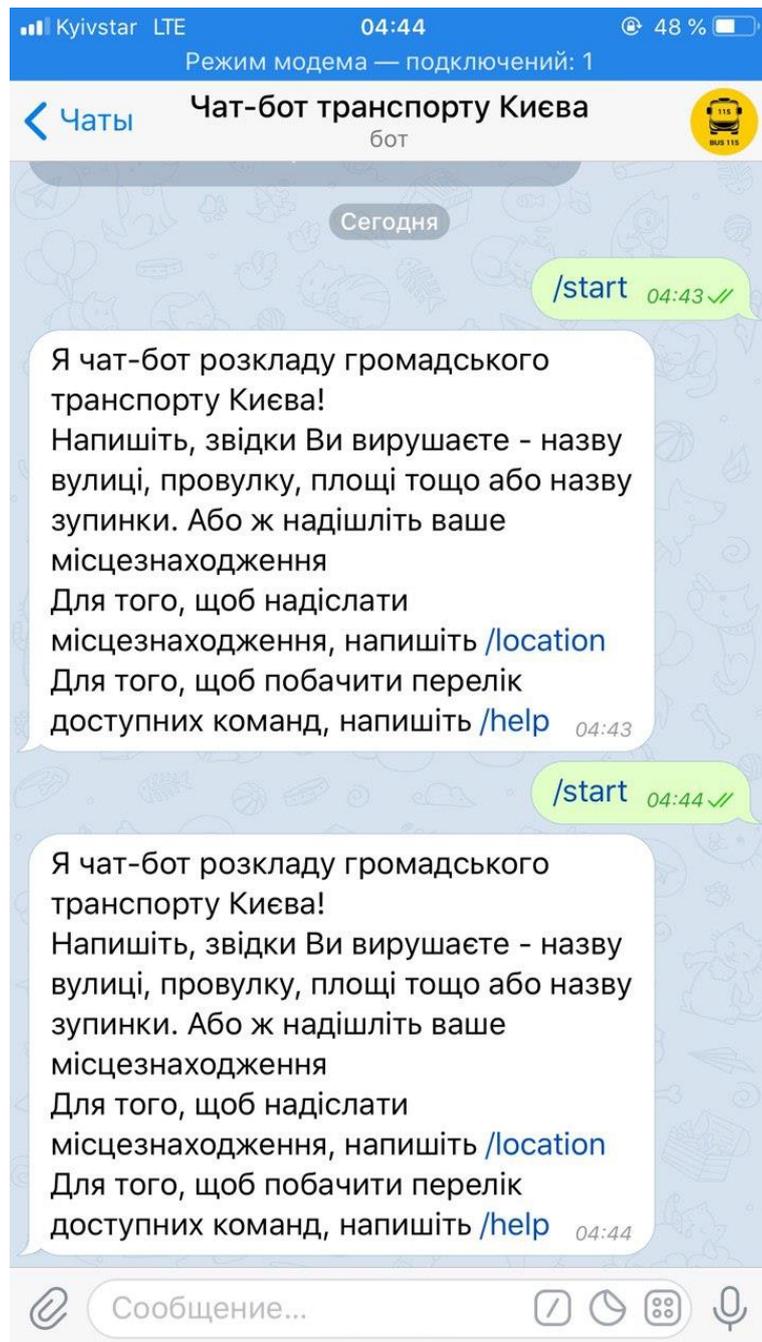


Рисунок 8.1 – Начало работы с ботом

После этого у нас появляется возможность отправить свое местонахождение, нажав кнопку Location или отправив команду /location. В свою очередь, бот в целях безопасности запросит разрешение на получение информации о местонахождении пользователя (рисунок 8.2).

Изм.	Письмо	№ докум.	Подпись	Дата

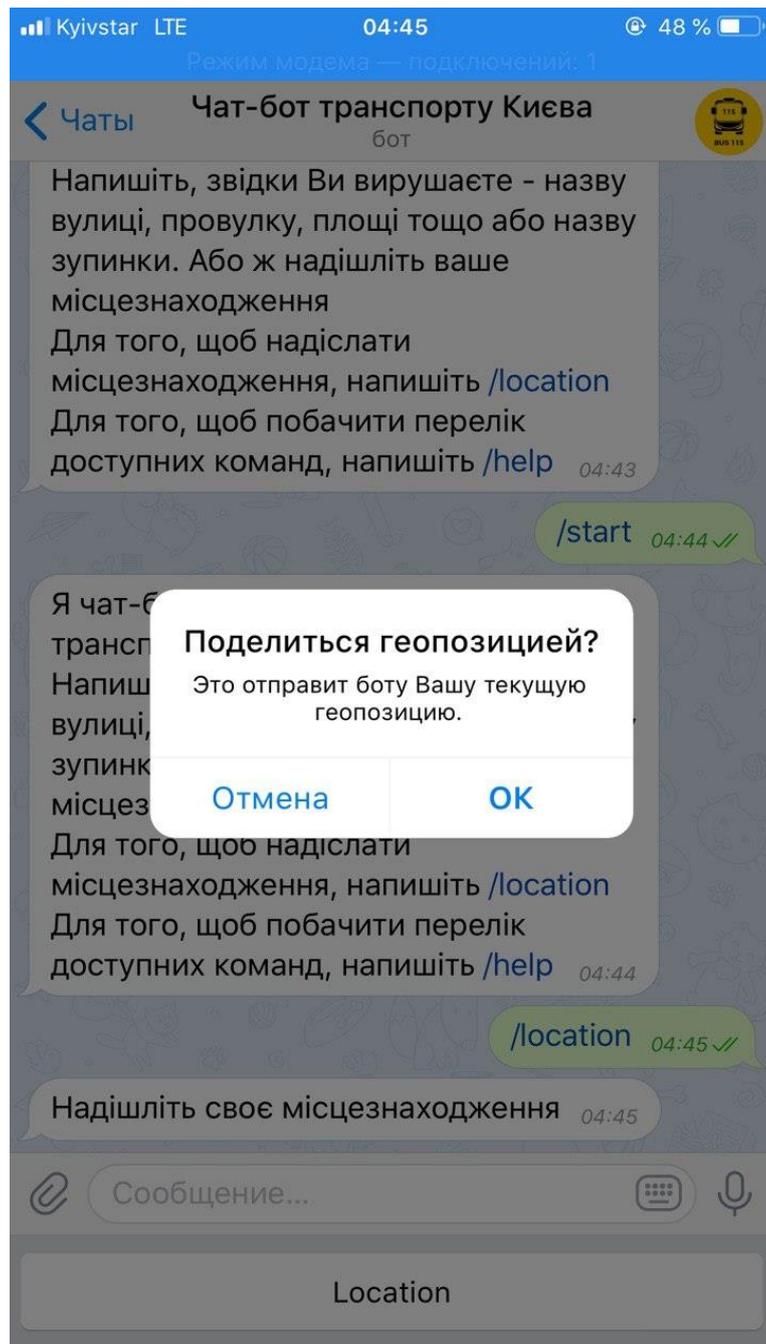


Рисунок 8.2 – Запрос на доступ к геопозиции

Как только пользователь согласится на получение ботом геопозиции, бот автоматически будет искать все возможные остановки общественного транспорта в радиусе 500 метров.

Введя адрес, который нужен пользователю, бот будет искать все возможные остановки общественного транспорта и отправит все найденные варианты в чат (рисунок 8.3).

Изм.	Письмо	№ докум.	Подпись	Дата

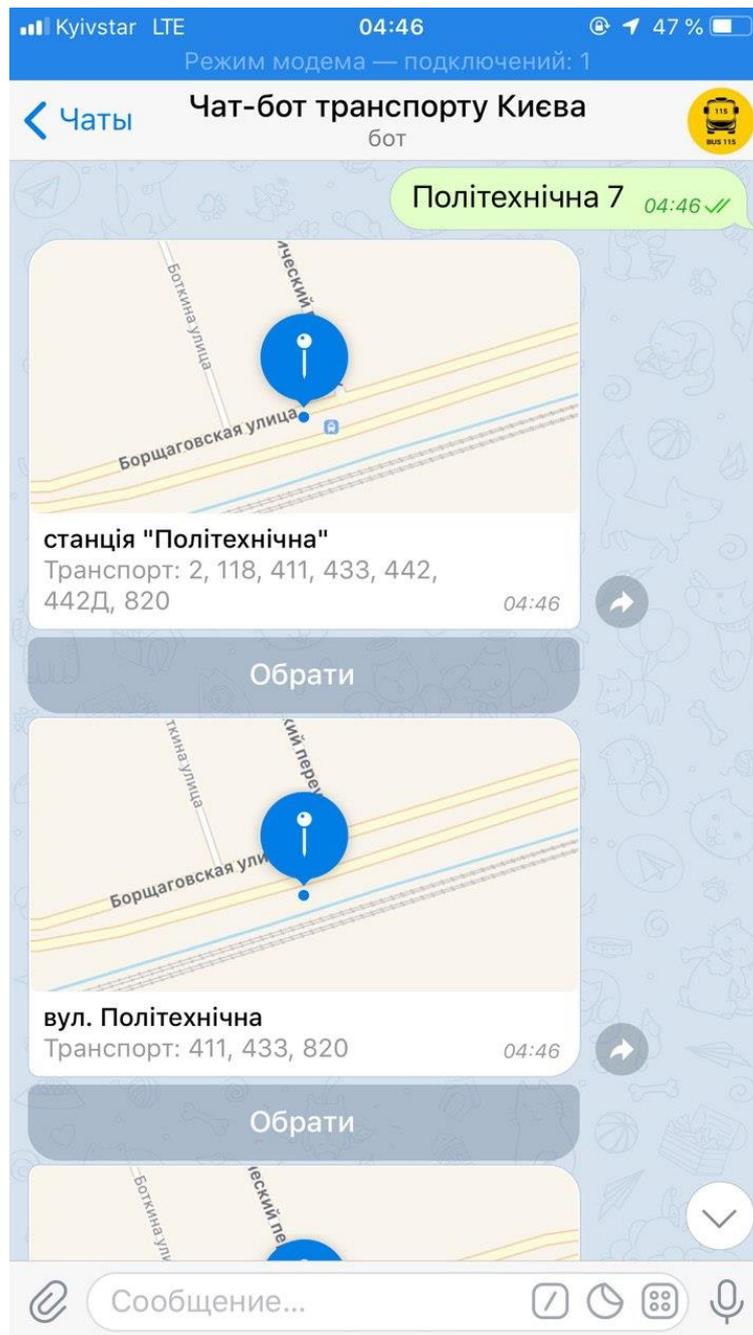


Рисунок 8.3 – Ответ бота по запросу

Также бот предоставляет возможность заранее видеть, какой общественный транспорт останавливается на той или иной остановке.

При выборе остановки общественного транспорта мы получаем информацию об ожидаемом времени прибытия общественного транспорта (рисунок 8.4).

Изм.	Письмо	№ докум.	Подпись	Дата

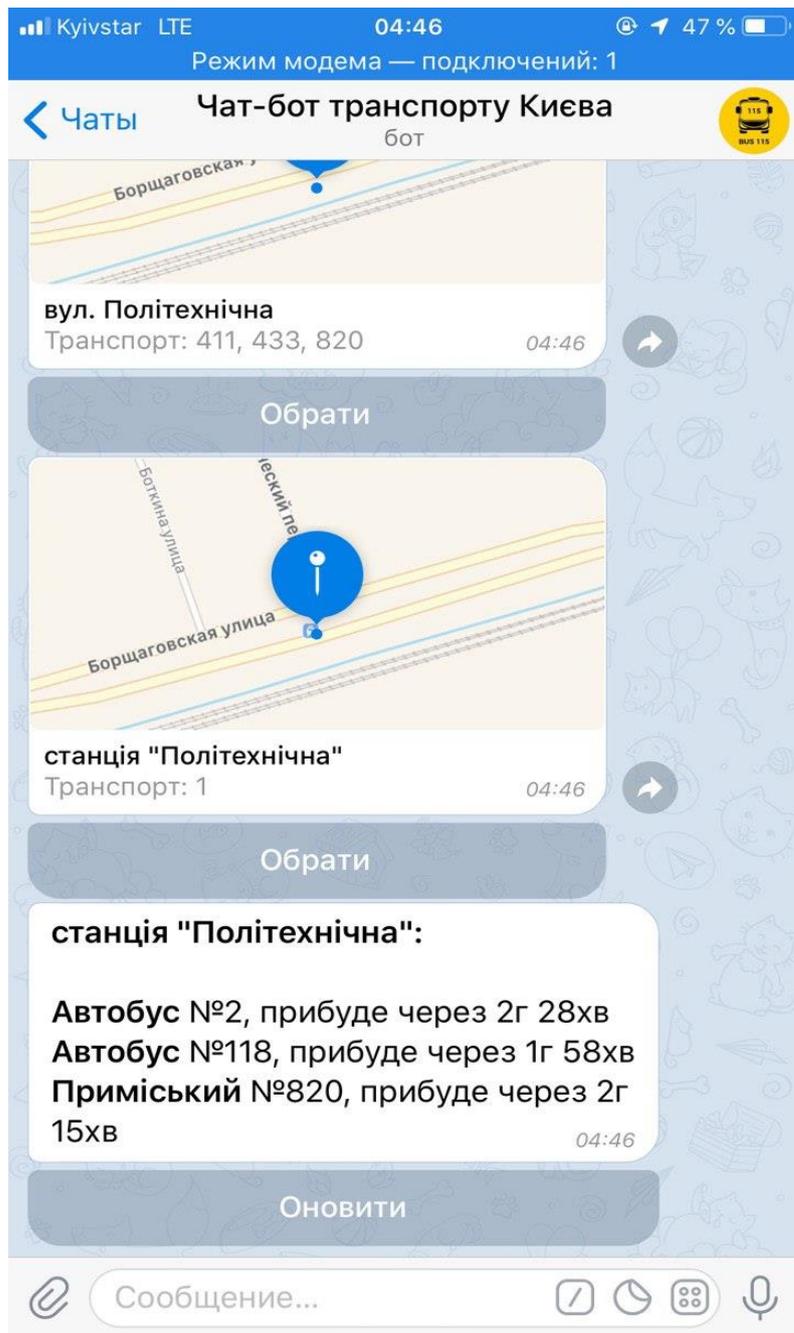


Рисунок 8.4 – Подробная информация об остановке

Изм.	Письмо	№ докум.	Подпись	Дата

9 ОПИСАНИЕ СОБСТВЕННОЙ РЕАЛИЗАЦИИ

В соответствии с заданием дипломного проекта и в соответствии с рассмотренными подходами было решено с помощью средств TeleBot, CherryPy и Requests на языке программирования Python создать телеграмм-бот, который будет предоставлять необходимую информацию.

Основные аспекты, которые планируется реализовать в ходе выполнения дипломного проекта:

- а) получение запроса пользователя; б) обработка запроса пользователя;
- в) при необходимости уточнить некоторые аспекты запроса;
- г) в соответствии с обработанным запросом выдать необходимую информацию пользователю;
- д) предусмотреть реакцию бота на непредвиденные запросы; Также

при возможности рассмотреть и реализовать следующие аспекты:

- а) создание базы данных общественного и остановок; б) получение координат местонахождения пользователя; в) предложить несколько вариантов остановок и маршрутов.

Более подробно опишем каждый из аспектов. Мы уверены, что пользователь захочет задать свой вопрос на естественном языке, или выбрать из предложенных нами вариантов, далее это сообщение будет передаваться в сервис, с помощью которого мы сможем получить необходимый нам контекст сообщения, затем эта информация будет передаваться в нашу программу, в которой мы будем определять что делать с этим сообщением, сразу предоставлять информацию, или нужно задать определенные уточняющие вопросы, процесс уточнения может повторяться до тех пор, пока мы не получим необходимую информацию для выдачи максимально полного ответа для пользователя, после завершения процесса получения запроса пользователя и его анализа, мы присылаем ему необходимую информацию.

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		57

Если пользователь ввел непредвиденный нами запрос или запрос, не касающийся нашей предметной области, то есть два варианта развития. Если запрос не касается темы, то будет отображаться сообщение о том, чтобы пользователь вводил вопросы только относительно транспорта. Если же пользователь ввел запрос о транспорте, но бот по какой-то причине либо не понял его, либо мы не смогли предоставить необходимую информацию по этому запросу, мы будем отвечать заранее подготовленными ответами.

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		58

ВЫВОДЫ

В ходе выполнения дипломной работы были проведены исследования предметной области, определены главные требования к системе и бизнес-процессы. Проанализированы требования к системе в целом, требования к функциям системы, программному и техническому обеспечению.

Было проведено исследование технологий для построения телеграмм-бота. В результате исследования были выбраны следующие языки программирования и технологии: Python, модуль TeleBot, модуль CherryPy, Requests, Telegram API и база данных MongoDB.

Результатом проведенного исследования явилось создание телеграмм-бота для построения маршрутов с использованием общественного транспорта города Киева. Система современна и удобна, удовлетворяет всем требованиям с точки зрения функциональности, юзабилити, логики.

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		59

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Подборка: 30 полезных ботов [Электронный ресурс] – Режим доступа: <https://ain.ua/2017/08/14/30-telegram-botov/>

2. Telegram Bot API [Электронный ресурс] – Режим доступа: <https://core.telegram.org/bots/api>

3. Самоучитель Python [Электронный ресурс] : Режим доступа: <https://pythonworld.ru/samouchitel-python>

4. Telegram FAQ [Электронный ресурс]. – Режим доступа: <https://telegram.org/faq>

5. Сайт Eway [Электронный ресурс]. – Режим доступа: <https://www.eway.in.ua/ru/cities/kyiv>

6. Telegram-bot[Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/442800/>

7. Телеграмм бот с помощью TeleBot [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/448310/>

8. Бот для Telegram на Python. Heroku сервер [Электронный ресурс]. – Режим доступа: <https://tproger.ru/translations/telegram-bot-create-and-deploy/>

9. CherryPy [Электронный ресурс]. – Режим доступа: <https://cherrypy.org/>

10. Создание бота. Telebot [Электронный ресурс]. – Режим доступа: <https://medium.com/@plug2up/>

[%D1%81%D0%BE%D0%B7%D0%B4%D0%B0%D0%B5%D0%BC-%D1%82%D0%B5%D0%BB%D0%B5%D0%B3%D1%80%D0%B0%D0%BC-%D0%B1%D0%BE%D1%82%D0%B0-2c8377d4d482](https://medium.com/@plug2up/%D1%81%D0%BE%D0%B7%D0%B4%D0%B0%D0%B5%D0%BC-%D1%82%D0%B5%D0%BB%D0%B5%D0%B3%D1%80%D0%B0%D0%BC-%D0%B1%D0%BE%D1%82%D0%B0-2c8377d4d482)

11. Python Requests [Электронный ресурс]. – Режим доступа: <https://2.python-requests.org/en/master/>

12. Руководство по работе с HTTP в Python. Библиотека requests [Электронный ресурс]. – Режим доступа: <https://khashtamov.com/ru/pythonrequests/>

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		60

13. Сайт Heroku [Электронный ресурс]. – Режим доступа:
<https://www.heroku.com/>
14. Heroku – Википедия [Электронный ресурс]. – Режим доступа:
<https://ru.wikipedia.org/wiki/Heroku>
15. Обзор протокола HTTP [Электронный ресурс]. – Режим доступа:
<https://developer.mozilla.org/ru/docs/Web/HTTP/Overview>
16. Очередь сообщений [Электронный ресурс]. – Режим доступа:
<https://aws.amazon.com/ru/message-queue/>
17. Synbot [Электронный ресурс]. – Режим доступа: https://ru.telegram-store.com/catalog/bots/synbot_/
18. REST API[Электронный ресурс]. – Режим доступа:
<https://habr.com/ru/post/351890/>
19. JSON[Электронный ресурс]. – Режим доступа:
<https://www.json.org/json-ru.html>
20. Тестирование телеграмм-бота [Электронный ресурс]. – Режим доступа:
<https://habr.com/ru/post/322816/>
21. What is Artificial Intelligence (AI) [Электронный ресурс]. – Режим доступа: <https://www.techopedia.com/definition/190/artificial-intelligence-ai>

					ИТ51.320БАК.002 ПО	Письмо
Изм.	Письмо	№ докум.	Подпись	Дата		61