**РЕФЕРАТ**

Текстова частина бакалаврcької роботи 57 с., 27 рисунків, 1 таблиця, 22 літературних джерела.

Ключові слова: Python, Telegram, API, Чат-бот, Криптовалюта

Об’єкт дослідження – моніторинг актуального курсу крiптовалют.

Предмет дослідження – методи та засоби моніторингу актуального курсу крiптовалют.

Мета роботи – покращення якості обслуговування користувачів шляхом розробки та впровадження чат-бота для моніторингу курсу криптовалют мовою програмування python у месенджері "Telegram".

Для реалізації поставленої мети потрібно вирішити наступні завдання:

1. Проаналізувати технічні засоби, що використовуються для розробки та обрати необхідні для створення **телеграм бота для моніторингу курсу крiптовалют**;

2. Розробити вимоги до застосунку на основі аналізу переваг та недоліків існуючих додатків;

3. Спроектувати та розробити систему на основі аналізу потреб користувачів;

4**.**  Провести тестування створеного телеграм бота.

Практичне значення отриманих результатів: Даний продукт може бути використаний для моніторингу актуального курсу кріптовалют з метою інтеграції у комерційні проекти.

Застосунок розроблений з використанням мови програмування Python, текстового редактору SublimeText, месенджеру Telegram та бібліотек pycoingecko та py\_currency\_converter для Python.

Галузь використання – моніторинг курсу кріптовалют.

**ABSTRACT**

Text part of the bachelor's thesis 55 pages, 27 figures, 1 table, 22 literary sources.

Keywords: Python, Telegram, API, Chatbot, Cryptocurrency

The object of research is the monitoring of the current cryptocurrency rate.

Subject of research - methods and means of monitoring the current cryptocurrency rate.

The purpose of the work is to improve the quality of customer service by developing and implementing a chatbot to monitor the cryptocurrency exchange rate in the python programming language in the "Telegram" messenger.

To achieve this goal you need to solve the following tasks:

1. Analyze the technical means used for the development and select the necessary to create a telegram bot to monitor the cryptocurrency rate;

2. Develop application requirements based on an analysis of the advantages and disadvantages of existing applications;

3. Design and develop a system based on user needs analysis;

4. Test the created telegram bot.

Practical significance of the obtained results: This product can be used to monitor the current cryptocurrency rate in order to integrate into commercial projects.

The application is developed using the Python programming language, SublimeText text editor, Telegram messenger and pycoingecko libraries and py\_currency\_converter for Python.

Area of ​​use - cryptocurrency exchange rate monitoring.

**ЗМІСТ**

**ПЕРЕЛІК СКОРОЧЕНЬ………………………………………………………8**

**ВСТУП………………………………………………………………………..….8**

**1. ОГЛЯД І ПОРІВНЯЛЬНИЙ АНАЛІЗ ЗАСОБІВ ДЛЯ РОЗРОБКИ СЕРВІСУ………………………………………………………………………11**

**1.1. Загальні поняття чат-бота. Функції чат-бота. Класифікація чат-ботів...11**

**1.2. Переваги та застосування чат-ботів у бізнесі ……... ………………..….…10**

**1.2.1 Переваги використання месенджерів для бізнесу……………...…15**

**1.2.2 Застосування чат-ботів у бізнесі………………………………..……16**

**1.3 Вибір програмного забезпечення розробки чат-бота………………..…..….19**

**1.3.1 Статистика та аналіз мов програмування для розробки чату в telegram…………………………………………………………………..……19**

**1.3.2 Основні поняття та види мов програмування……………….……20**

**1.3.3 Найпопулярніші мови програмування………….................………..21**

**1.3.4 Порівняльний аналіз мов програмування…………….......………..22**

**1.3.5 Вибір текстового редактора для написання коду………...………..27**

**1.4 Результати та висновки………………………………..………………………..31**

**2**. **ОПИС МЕТОДОЛОГІЇ РОЗРОБКИ СЕРВІСУ………………………....32**

**2.1 Необхідні модулі для створення сервісу…………………………...………….32**

**2.2 Python як засіб розробки чат-бота…………....………………………………..33**

**2.2.1 Функції………....………………………………………………………..33**

**2.2.2 Умовні оператори та цикли…………………..………………………34**

**2.2.3 Конструкції……....……………………………………………………..35**

**2.2.4 Модулі……………………….…………………………………………..36**

**2.3 Принцип отримання актуальних даних з курсу криптовалют……..……..36**

**2.4 Проектування UML діаграми варіантів використання…………………....38**

**2.5Результати та висновки……..………………………………………………….39**

**3. ПРАКТИЧНА РЕАЛІЗАЦІЯ СЕРВІСУ ТЕЛЕГРАМ БОТА ДЛЯ МОНІТОРИНГА КУРСУ КРИПТОВАЛЮТ………………..………….….39**

**3.1 Опис процесів TO-BE…………………………………………………………….39**

**3.2 Структура проекту………………....……………………………………………..41**

**3.3 Ризики проекту……………………...…………………………………………….43**

**3.4 Практична реалізація чат-бота мовою програмування Pyhton…….......…..45**

**3.4.1 Отримання токену для функціонування боту…......................……..45**

**3.4.2 Написання програмного коду роботи бота мовою Python….....…..46**

**3.4.3 Результати роботи бота…………….............………………………….48**

**ВИСНОВОК……………………………………………..………………………50**

**СПИСОК ЛІТЕРАТУРИ…………………………………………...……….....51**

**ДОДАТКИ……..……………………………………………………………...…55**

**ДОДАТОК А………………………………………………………………………….57**

**ДОДАТОК Б……………………………………………………….………………....57**

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

ПК – персональний комп’ютер

Python – мова програмування

Telegram– кросплатформова система миттєвого обміну повідомленнями

API - опис методів, якими одна комп'ютерна програма може взаємодіяти з іншою програмою.

Token - унікальний ідентифікатор програми, що вимагає доступ до вашої служби..

**ВСТУП**

Ринкові умови, що постійно змінюються, висока швидкість прийняття рішень, багатозадачність в управлінні активами та необхідність зниження ризиків вимагають сучасних підходів до організації підприємницької діяльності. Рішення все більш складного внутрішнього та зовнішнього середовища підприємства полягає у комплексній автоматизації бізнес-процесів. Це дозволяє вивільнити цінні ресурси для стратегічного планування та концентрації управління у ключових сферах діяльності підприємства.

Необхідність автоматизації інформаційних процесів обумовлена збільшенням обсягу інформації в інформаційній системі (ІС) організації, необхідністю прискорення та використання більш складних методів їх обробки.

Автоматизація бізнесу – це частковий або повний переклад стереотипних операцій та бізнес-завдань під управління спеціалізованою інформаційної системи або складного апаратного та програмного забезпечення. Внаслідок чого відбувається вивільнення людських та фінансових ресурсів для підвищення продуктивності праці та ефективності стратегічного керування.

аналізу та підвищення точності обліку та звітності інформації;

Мета роботи: покращення якості обслуговування клієнтів шляхом розробки та впровадження чат-бота, що автоматизує моніторинг курсу криптовалют мовою програмування python у месенджері "Telegram". Для виконання поставленої мети необхідне вирішення наступних задач:

- вивчити поняття чат-бота та його функцій;

- розглянути переваги використання месенджерів для бізнесу;

- розглянути мови програмування та вибрати мову для розробки чат-бота;

- розробити архітектуру програмного продукту;

- описати процеси для впровадження чат-бота;

- розробити ІТ-проект із розробки чат-бота;

- продемонструвати концепцію розробки;

**1. ОГЛЯД І ПОРІВНЯЛЬНИЙ АНАЛІЗ ЗАСОБІВ ДЛЯ**

**РОЗРОБКИ СЕРВІСУ**

**1.1 Загальні поняття чат-бота. Функції чат-бота. Класифікація чат-ботів**

Чат-боти - це спеціальні акаунти, за якими не закріплена будь-яка людина, а повідомлення, надіслані з них або на них, обробляються зовнішньою системою. Крім того, для користувача спілкування з ботом виглядає як просте листування з справжньою людиною .

Чат-бот — це розумна програма, яка живе у месенджерах та виконує різні функції.

Функції чат-бота:

1) Підтримка клієнтів

Чат-бот допоможе замінити незручний FAQ на сайті, який іноді не одночасно можна побачити, чи зможе відповісти на типові питання клієнта. Бот може працювати 24 години на добу та розвантажить ваших співробітників.

2) Клієнтський сервіс

За допомогою чат-бота можна робити покупки та бронювати послуги. В роздрібній торгівлі, з постійним розширенням асортименту, складніше шукати конкретні товари. Після невеликого аналізу бот зрозуміє, що цікавить клієнта і відправить пряме посилання.

3) Маркетинг

Чат-бот - це ще один маркетинговий інструмент, який допоможе розповсюджувати контент, підтримувати лояльність клієнтів та збирати аналітику. За допомогою нього можна робити розсилки, інформувати клієнтів про акції, збирати коментарі про товари або послуги, якість обслуговування.

4) Робота всередині компанії

Чат-боти допомагають оптимізувати такі процеси як: бронювання переговорів, інформування працівників про дати відпустки, розклад корпоративного транспорту, терміни зарплати та багато іншого.

5) Рекрутинг

6) Функціональність просунутих ботів - це первинний збір інформації про кандидатів.На основі співбесіди з чат-ботом менеджер з персоналу вирішує, яких кандидатів слід запросити на живу співбесіду, хто має пройти тестове завдання, а кому слід відмовити у реченні про роботу. Існує кілька варіантів класифікації чат-ботів, але проаналізувавши їх усі, виділимо два види: бізнес-класифікація чат-бот додатків та класифікація чат-ботів за технічним типом.

Діаграма бізнес-класифікації представлена на рисунку 1.1.



Рисунок 1.1 – Бізнес-класифікація чат-бот додатків

Розглянемо кожен тип докладніше:

* Розмовні – створені для спілкування, дуже схожі на спілкування зі звичайною людиною, яке має конкретну мету.
* Асистенти – виходячи з конкретних цілей, з користувальницьких відповідей отримують необхідні дані.
* Q&A(питання-відповідь) – принцип роботи: одне питання – одна відповідь.

Класифікація за технічним типом представлена ​​на діаграмі (Рисунок 1.2).



Рисунок 1.2 – Технічна класифікація чат-ботів

Розглянемо кожен тип докладніше:

* Засновані на бізнес-правилах. У такому типі розмова людини та бота заздалегідь продумана розробником і має деревоподібну структуру. Завдяки великій кількості кнопок людина приходить до певного шляху. Запитань з відповіддю у вільній формі у такому типі не існує.
* Засновані на штучному інтелекті. Повністю відрізняються від першого типу, не мають визначеної структури. Шлях розмови визначено неявним чином на основі тестованих даних, які використовувалися для навчання моделі машинного навчання. Такі боти повинні мати великий обсяг даних для якісної роботи.
* Гібридні. Цей тип чат-ботів використовує у собі взаємодію першого і другого типу, тобто розмова з користувачем заздалегідь певного типу, але використовується штучний інтелект для визначення намірів користувача, та вилучення даних їх листування.

**1.2 Переваги та застосування чат-ботів у бізнесі**

Згідно з дослідженням LiveWorld, три роки тому 58% маркетологів не хотіли вкладати кошти в месенджери та чат-ботів. Водночас аналітики зробили більш оптимістичні прогнози щодо автоматизованих процесів та штучного інтелекту - 85% назвали месенджер-маркетинг новим стандартом для бізнесу. І вони мали рацію.

Соціальні мережі процвітають вже майже 10 років: вони стали не лише платформою для спілкування між людьми, а й для ведення бізнесу. Багато компанії зобов'язані своєю популярністю та фінансовим успіхом соціальним мережам.

Кардинальне зрушення відбулося у 2018 році. Близький родич соціальних мереж - Messenger - випередив соціальні мережі за кількістю користувачів. За оцінками The Economist, люди витрачають більше двох годин на день на програми обміну повідомленнями. Така активна аудиторія могла впливати лише на нові тенденції у маркетингу. Компанії використовували не тільки месенджери як повноцінну платформа для спілкування з клієнтами та рекламними послугами, а й для прямих продаж.

**1.2.1 Переваги використання месенджерів для бізнесу**

Переваги месенджерів:

1) Конфіденційність.

Соціальні мережі за своєю природою є відкритими платформами. Через них бізнесу легко транслювати свої повідомлення, спілкуватися з аудиторією, яка вже знайома з брендом, та залучати нову. Але соціальні мережі не можуть похвалитися надійним зберіганням конфіденційної інформації. У 2019 році Facebook заплатив самий великий в історії штраф за витік даних - п'ять мільярдів доларів.

Месенджери WhatsApp, Facebook Messenger, Viber та Telegram – це канали, якими інформація передається безпосередньо одержувачу. Крім того, більшість програм обміну миттєвими повідомленнями мають додаткову наскрізну систему шифрування, що також робить цю платформу безпечною для банківського сектора.

2) Керування репутацією.

Спілкування тет-а-тет через месенджери набагато краще. По перше, проблеми ваших клієнтів не стануть предметом обговорення для всієї аудиторії Facebook та ЗМІ. В особистому листуванні можна виконувати запит більш якісно та ймовірність, що клієнт залишиться задоволеним зростає. По-друге, при листуванні в месенджері можна виявити як позитивні якості компанії, і негативні.

3) Зручне середовище для зв'язку.

В основі месенджерів — асинхронність, швидкість та всі ваші діалоги в одному місці. Поєднання всіх цих властивостей робить їх основним цифровий засіб комунікації. Згідно з новим дослідженням, глобальні продажі штучного інтелекту зростуть до 126 мільярдів доларів до 2025 року порівняно з 10,1 мільярдами доларів у 2018 році. Тепер штучний інтелект бачить своє продовження у чатах. Враховуючи те, як бренди впроваджують віртуальні помічників, очікується, що вони стануть основними інвестиціями обслуговування клієнтів протягом наступних кількох років.

**1.2.2 Застосування чат-ботів у бізнесі**

1) Бот Comedy Club (рисунок 1.3)

Платформа: Viber.

Призначення: дистрибуція відеоконтенту на запит.

Функція: бот дає передплатникам можливість переглянути відеоконтент в один клік. Глядач отримує доступ до відеобібліотеки Comedy Club через 3 кнопки: «Новий випуск», який з'являється відразу після його ефіру на ТНТ, «Поточний сезон» та "Вибір дня".

Мета проекту: спростити життя глядача, про монетизацію поки що мова не йде.

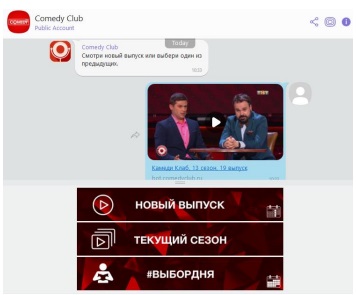


Рисунок 1.3 – Чат-бот компанії Comedy Club

2) Бот каршерингу YouDrive (рисунок 1.4)

Платформа: Telegram.

Бот на ⅔ знизив кількість дзвінків до компанії.

Мета робота: зробити процес повідомлення про проблеми з автомобілем або прокатом простим та зручним для клієнта, а також підвищити швидкість обробки викликів працівниками служби, що скорочує час очікування рішення.

Функції:

* Бот пропонує готові моментальні рішення різних питань: від «як завести автомобіль» до «що робити, якщо я потрапив у ДТП».
* розподіляє звернення користувачів за відповідальними співробітникам: кожен тип звернення надсилається до відповідного відділу для оперативного вирішення проблеми.

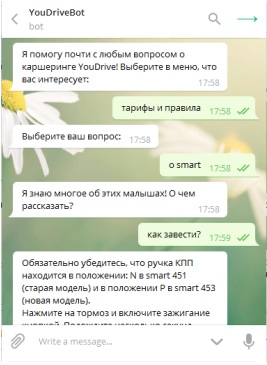


Рисунок 1.4 – Чат-бот каршерингу YouDrive

3) Бот таксі «Максим» (рисунок 1.5)

Платформа: Telegram.

Призначення: замовлення машини без дзвінків.

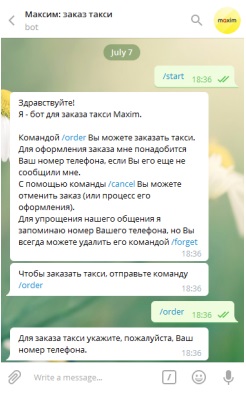


Рисунок 1.5 - Чат-бот таксі "Максим"

4) Бот компанії Aviasales

Платформи: Telegram, Slack.

Призначення: пошук квитків зі знижками.

Ціль: збільшити конверсію продажу квитків.

Чат-боти Telegram збільшують продажі. Завдяки їм у короткий термін можна отримати цільову аудиторію. При правильному використанні бота він допоможе у продажах та надасть необхідну інформацію покупцю. Як показує практика, використання чат-бота для підтримки клієнтів може зменшити до 40% часу консультантів в онлайн-чатах. Крім того, до половини звернень до робота відбувається у неробочий час. На відміну від живої людини, бот здатний одночасно давати відповіді кільком користувачам. Більше того, відповідь надається миттєво. Великою перевагою чат-ботів є кросплатформність. Готового бота не складно адаптувати до інших платформ. Тому ви можете одразу замовити бота для Telegram, Facebook, Messenger та WhatsApp, не розробляючи їх окремо.

Виходячи з перерахованого та розглянутого вище можна зробити висновок, що популярність чат-ботів набирає обертів та компанії, що прагнуть до збільшення ефективності свого бізнесу, все частіше замовляють розробникам нових чат-ботів.

**1.3 Вибір програмного забезпечення розробки чат-бота**

**1.3.1 Статистика та аналіз мов програмування для розробки чату в telegram**

У сучасному світі дуже багато зав'язано на IT-технологіях, практично в будь-якій компанії працівники використовують різні програми для ефективної та якісної роботи, світ не стоїть на місці, а розвивається і тематика цієї статті обумовлена ​​тим, що в рамках дипломної роботи є ідея написання чат-бота в месенджері Telegram, який буде надавати доступ до моніторингу курсу криптовалют. Для написання чат-бота необхідно вибрати мову програмування від якої залежатиме швидкість написання коду, можливості мови тощо.

**1.3.2 Основні поняття та види мов програмування**

Мова програмування - це набір правил, які визначають як виглядає написана комп'ютерна програма і що комп'ютер може робити під її контролем. Програма – це код, написаний відповідно до правилами цієї мови програмування. Код, із якого складається програма називається «вихідним кодом». Мови програмування – це формальні штучні мови. Як і природні мови, вони мають алфавіт, словниковий запас, граматику та синтаксис, і навіть семантику.

Алфавіт – дозволений для використання набір символів, за допомогою якого можуть бути утворені слова та величини даної мови.

Синтаксис - система правил, що визначають допустимі конструкції

мови програмування з літер алфавіту.

Семантика - система правил однозначного тлумачення кожної мовної конструкції, що дозволяють виробляти процес обробки даних

Всі мови програмування поділяються на два види – мови низького та високого рівня :

* Мови низького рівня – це спосіб написання комп'ютерних інструкцій апаратною мовою, тобто в машинних кодах (у вигляді послідовності нулів та одиниць). Мови низького рівня жорстко орієнтовані на конкретний тип обладнання (система управління процесором, кожен тип процесора має свій машинний код).
* Мови високого рівня – це мови програмування, які дозволяють записувати програми у зручній для людини формі. Ці мови орієнтовані не так на систему інструкцій тієї чи іншої процесора, а на систему операторів (інструкцій), характерну для написання певного класу алгоритмів.

Мови високого рівня простіше у використанні, оскільки їхнє завдання - обслуговувати потреби програміста, а не визначати можливості комп'ютера. Програми, написані цими мовами, мають бути перекодовані - перекладені машинною мовою, щоб перед запуском програм комп'ютер міг їх зрозуміти. Тому системи програмування на Java включають або інтерпретатор мови, або компілятор. Мови низького рівня, близькі до машинної мови, дозволяють створювати програми, які працюють швидше та дозволяють більш ефективно використовувати ресурси комп'ютера.

**1.3.3 Найпопулярніші мови програмування**

На сьогоднішній день існує велика кількість різних мов програмування і в кожного з них своя сфера застосування, але все ж для проведення аналізу на вибір кращої мови для написання чат-бота, необхідно вибрати кілька найпопулярніших мов, щоб між ними проводити аналіз, тож у цьому розділі звернемося до статистики за популярністю мов. На малюнку 1.6 показаний рейтинг мов 2022 у комерційних розробках.

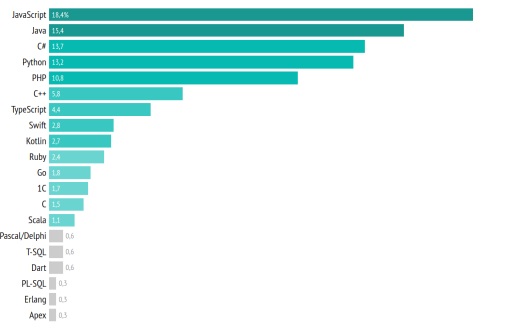


Рисунок 1.6 – Рейтинг мов програмування у комерційних проектах, %

За цими даними можна зробити висновок, що JavaScript значно випереджає Java і є найпопулярнішою мовою програмування. У п'ятірку найкращих мов увійшли також: C#, Python, PHP. На малюнку 1.7 можна подивитися, як змінювалися дані на протязі 2012-2022 років.

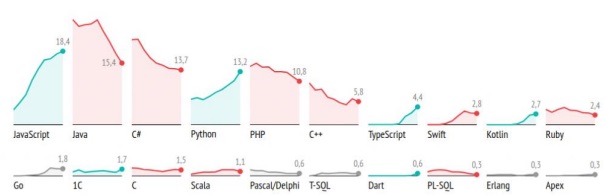


Рисунок 1.7 – Графік зміни популярності мов програмування, %

На малюнку 1.7 можна побачити, що популярність мови Java і C# сильно падає, а популярність таких мов як JavaScript, TypeScript та Python продовжує зростати.

**1.3.4 Порівняльний аналіз мов програмування**

Python — це високорівнева мова загального програмування призначення, яке також використовується для розробки веб-додатків. Мова націлена на підвищення продуктивності праці розробників та читабельності коду.

Python підтримує різні парадигми програмування: структурну, об'єктно-орієнтовану, функціональну, імперативну та аспектно-орієнтовану. Мова включає динамічну типізацію, автоматичне керування пам'яттю, повне самоспостереження, механізм обробки винятків, підтримку багатопоточних обчислень та практичні структури даних високого рівня.

Переваги Python :

* відкрита розробка;
* досить простий у вивченні, особливо на початковому етапі;
* особливості синтаксису стимулюють програміста писати код, що легко читається;
* надає засоби швидкого прототипування та динамічної семантики;
* має велику спільноту, позитивно налаштовану по відношенню до новачкам;
* безліч корисних бібліотек та розширень мови можна легко використовувати у своїх проектах завдяки гранично уніфікованим механізмом імпорту та програмним інтерфейсів;
* механізми модульності добре продумані та можуть бути легко використані;
* абсолютно все в Python є об'єктами в сенсі ООП, але при цьому об'єктний підхід не нав'язується програмісту.

Недоліки Python:

* не надто вдала підтримка багатопоточності;
* на Python створено не так вже й багато якісних програмних проектів у порівнянні з іншими універсальними мовами програмування, наприклад, з Java;
* відсутність комерційної підтримки засобів розробки (хоча ця ситуація з часом змінюється);
* початкова обмеженість коштів для роботи з базами даних;
* бенчмарки показують меншу продуктивність Python за порівняно з основними Java VM, що створює цій мові репутацію повільного.

JavaScript – мультипарадигменна мова програмування. Підтримує об'єктно-орієнтований, імперативний та функціональний стилі.

* Переваги JavaScript:
* Жоден сучасний браузер не обходиться без підтримки JavaScript.
* З використанням написаних на JavaScript плагінів та скриптів впорається навіть не фахівець.
* Корисні функціональні налаштування.
* Мова, що постійно вдосконалюється – зараз розробляється бета-варіація проекту, JavaScript2.
* Взаємодія з додатком може здійснюватися навіть через текстові редактори – Microsoft Office та Open Office.
* Перспектива використання мови у процесі навчання програмування та інформатики.

Недоліки JavaScript:

* Знижений рівень безпеки через повсюдний та вільного доступу до вихідних кодів найпопулярніших скриптів.
* Безліч дрібних дратівливих помилок на кожному етапі роботи. Більшість їх легко виправляється, але їх наявність дозволяє вважати цю мову менш професійною, порівняно з іншими.
* Повсюдне поширення. Своєрідним недоліком можна вважати той факт, що частина програм, що активно використовуються (особливо додатків) перестануть існувати за відсутності мови, оскільки повністю базуються на ній.

C# - мова програмування, що поєднує об'єктно-орієнтовані та контекстно-орієнтовані концепції.

Переваги:

* для малих підприємств та деяких окремих розробників безкоштовні інструменти включають Visual Studio, Azure Cloud, Windows Server, Parallels Desktop для Mac Pro та багато інших;
* велика кількість синтаксичних конструкцій, розроблених для кращого розуміння написання коду;
* дуже простий у вивченні;
* після покупки Xamarin на C# ви можете писати програми та програми для операційних систем, таких як iOS, Android, MacOS та Linux;
* є ціла спільнота з досвідчених програмістів.

Недоліки:

* пріоритетна орієнтованість на Windows платформу;
* мова безкоштовна тільки для невеликих фірм, індивідуальних програмістів, стартапів та учнів. Великої компанії покупка ліцензійної версії цієї мови коштуватиме круглу суму;
* у мові залишилася можливість використання оператора безперечного переходу.

Представимо синтаксис мов по написанню рівнозначного за результатом коду на рисунках 1.8, 1.9 і 1.10.



Рисунок 1.8 – Приклад коду на Java



Рисунок 1.9 – Приклад коду на С



Рисунок 1.10 – Приклад коду на Python

По малюнках видно, що, коли сімейство Java і C вимагають 4-5 рядків коду, Python використовує лише одну. Щоб зробити вибір мови програмування для чат-бота, потрібно зрозуміти, що таке чат-бот. Чат-бот – це програма, яка з'ясовує потреби користувачів, а потім допомагає задовольнити їх.

За статистикою та особистим аналізом, ,було з’ясовано, що найчастіше в якості мови програмування для чат-бота вибирають Python. Наведемо кілька прикладів переваг створення бота на Python:

* Ваші можливості практично безмежні.
* Економія грошей.
* Отримання навичок, які можуть стати в нагоді – як мінімум, ви можете заробляти на створенні "ботів".

Недоліки:

* Витрата часу та сил на вивчення мови програмування.

Підіб'ємо підсумки, у кожної мови програмування є свої плюси і мінуси, кожна мова хороша для конкретних цілей, наприклад, якщо створювати сайт, то за описаними критеріями потрібно вибрати JavaScript, а для написання чатбота в месенджері Telegram більше підійде Python так як він більш простий у написанні, має багато документації і має великий вибір бібліотек, які допоможуть створити якісний програмний продукт.

**1.3.5 Вибір текстового редактора для написання коду**

Для написання коду нам потрібний текстовий редактор – це незалежна комп'ютерна програма або компонент програмного комплексу, призначений для створення та зміни текстових даних та текстових файлів. Редактор відіграє важливу роль у кожній галузі роботи розробника. Код написаний, налагоджений та виконаний з використанням текстового редактора.

Вибір ідеального редактора для роботи - складне завдання, яке включає: тестування, особисті переваги і остаточне рішення. Перш ніж зробити вибір, нам потрібно проаналізувати деякі з них.

Sublime Text editor (рисунок 1.11) - один з найкращих текстових редакторів на сьогоднішній день. Це відмінна альтернатива потужної IDE, вона легка та виконує свою роботу з високою ефективністю та точністю.

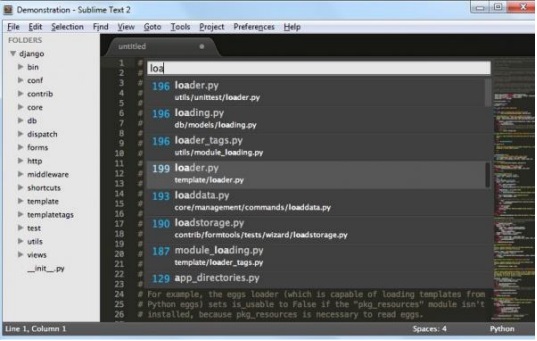


Рисунок 1.11 – Sublime Text editor

Sublime Text має багато можливостей, але, як і будь-який інструмент він не є досконалим. Давайте розглянемо плюси та мінуси використання Sublime Text editor.

Переваги:

- Приємний, легкий, мінімалістичний інтерфейс.

- Дуже гнучко налаштовується. Множинне виділення.

- Можливість створення будь-яких сніпетів та вставки їх хоч по гарячим клавішам, хоч за літерними скороченнями.

- Можливість призначення гарячих клавіш абсолютно на будь-яку дію

- У сніпетах можна задати, де буде курсор при вставці, задати плейсхолдери і перемикання в потрібні ділянки сніпету клавішею Tab.

- Наявність мінікарти коду для зручного переміщення.

- Можливість відображення прихованих символів.

Недоліки:

- Sublime text платний. Враховуючи, що на ринку багато ефективних безкоштовних редакторів, кожен може вибрати альтернативу.

- Час завантаження вищий у порівнянні з Notepad++.

- Загалом, Sublime text – чудовий редактор для роботи. Він може бути використаний безкоштовно з нескінченним пробним періодом.

Visual Studio Code (малюнок 1.12) – це продукт корпорації Microsoft, розроблений для того, щоб дозволити розробникам писати код без завантаження масивної Visual Studio (3 ГБ+). Visual Studio Code - це простий редактор з відкритим вихідним кодом, який однаково добре працює у Windows, OS X та Linux. Основні функції коду Visual Studio включають підтримку понад 30 мов, автозаповнення, просту навігацію тощо. Він також включає засоби налагодження та Git для полегшення розробки.

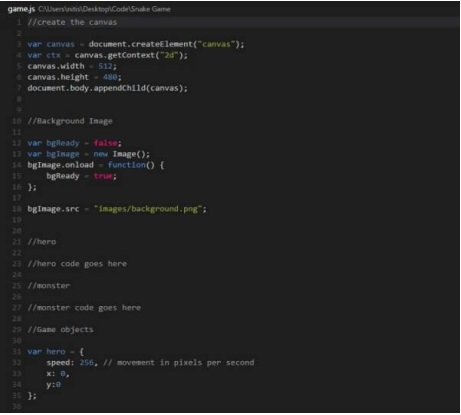


Рисунок 1.12 – Visual Studio Code

Переваги:

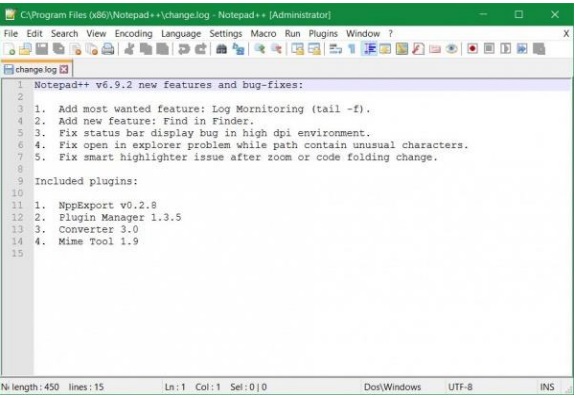
* Підтримка більш ніж 30 мов, а також основні мови Microsoft, такі як ASP.NET, C# і т.д.
* Невеликий розмір гарантує швидку установку та використання.

Недоліки:

* Підтримка розширень потребує покращення.
* Оновлення на Linux може бути непростим завданням.

Visual Studio Code – чудовий вибір для програмістів, які не хочуть завантажувати та використовувати громіздкі інтегровані середовища розробки.

Notepad ++ (Рисунок 1.13) - це редактор з відкритим вихідним кодом. Він схожий на блокнот, який підтримує багато мов програмування. Найбільшою перевагою Notepad ++ є те, що він може обробляти великі файли без тривалих затримок чи збоїв. Цей редактор дуже швидкий і простий у використанні.



Малюнок 1.13 – Notepad++

Переваги:

* + підтримка великої кількості кодувань;
  + підсвічування синтаксису;
  + паралельне редагування документів;
  + порівняння документів;
  + пошук та автозаміна за регулярними виразами;
  + робота з файлами на сервері FTP.

Недоліки:

* + не найкращий інтерфейс користувача;
  + надто простий.

Notepad ++ відмінно підійде тим, хто задоволений простим інтерфейсом та набором функцій. Він допоможе вам швидко та легко створювати нові продукти.

Для написання сервісу було обрано SublimeText, враховуючи все переваги та недоліки, виявився найзручнішим для роботи.

**1.4 Результати та висновки**

У цьому розділі було розглянуто загальні поняття чат-бота, показано функції, наведена класифікація всіх існуючих на даний момент чатботів. Також були показані переваги використання месенджерів у бізнесі. Також було проведено вибір програмного забезпечення для реалізації чат-бота, за результатами якого були обрані такі засоби: мова програмування Python, текстовий редактор Sublime Text, месенджер Telegram.

**2. ОПИС МЕТОДОЛОГІЇ РОЗРОБКИ СЕРВІСУ**

**2.1 Необхідні модулі для створення сервісу**

З першого розділу ми з'ясували, що мова програмування Python добре підійде для написання чат-бота для месенджера Telegram Розглянемо бібліотеки даної мови, необхідні для написання чат-бота.

Модуль python-telegram-bot забезпечує чистий інтерфейс Python для Telegram Bot API. Він сумісний із версіями Python 3.6+. Пакет python-telegram-bot може працювати з PyPy3 (офіційно не підтримується). API бота надається через клас telegram.Bot. Методи, визначені в telegram.Bot, є еквівалентами у вигляді методів snake\_case, описаних в офіційній документації Telegram Bot API. Для зручності також доступні точні назви методів у вигляді camelCase, зазначені в документації Telegram. Так, наприклад, виклик telegram.Bot.send\_message збігається із викликом методу telegram.Bot.sendMessage. Усі класи об'єктів Telegram Bot API розташовані в основному модулі пакета telegram, наприклад, клас об'єкта Message доступний як telegram.Message.

Щоб створити токен доступу, необхідно поспілкуватися з @BotFather і виконати кілька простих кроків, описаних у розділі Команди та оповіщення @BotFather в Telegram.

Крім чистої реалізації API ця бібліотека містить ряд високорівневих класів, які роблять розробку роботів простою і зрозумілою. Ці класи містяться у модулі telegram.ext.

Модуль pycoingecko надає класи для обробки даних про актуальний курс криптовалюти. Звернення до модулю виконується за допомогою команди cg.get\_price.

Модуль py\_currency\_converter надає класи для конвертації однієї валюти в іншу. Звернення до модулю виконується за допомогою команди convert.

JSON (JavaScript Object Notation) - простий формат обміну даними,

заснований на підмножині синтаксису JavaScript. Модуль json дозволяє

кодувати та декодувати дані у зручному форматі.

**2.2 Python як засіб розробки чат-бота**

Для розробки чат-боту засобами мови Python нам потрібно ознайомитись з базовими термінами цієї мови.

**2.2.1 Функції**

Функція в Python – це об'єкт, який приймає аргументи та повертає значення. Як правило, функція визначається за допомогою оператора def. Найпростіша функція показана малюнку 2.1.



Рисунок 2.1 – Функція

Функція може приймати будь-яку кількість аргументів чи не приймати їх взагалі. Часто зустрічаються функції з будь-якою кількістю аргументів, функції з позиційними та іменованими аргументами, обов'язкові та необов'язкові (рисунок 2.2).



Рисунок 2.2 – Аргументи функції

**2.2.2 Умовні оператори та цикли**

Умовний оператор if-elif-else (іноді званий оператором розгалуження) є основним інструментом вибору Python. Простіше кажучи, він вибирає, яку дію зробити, залежно від значення змінних у момент перевірки умови.

Синтаксис: спочатку записується частина if з умовним виразом, потім одна або кілька необов'язкових частин elif і, нарешті, необов'язкова частина else. Загальна форма умовного листа виглядає так (рисунок 2.3):

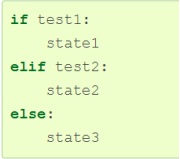


Рисунок 2.3 – Синтаксис умовного оператора

If While - хоча це один з найуніверсальніших циклів у Python, він досить повільний. Виконує тіло циклу доти, доки умова циклу істинно (рисунок 2.4).

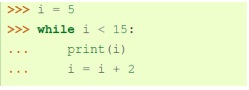


Рисунок 2.4 – Синтаксис умовного оператора While

Цикл for вже трохи складніший, трохи менш універсальний, але працює набагато швидше, ніж цикл while. Цей цикл проходить через будь-який повторюваний об'єкт (наприклад, рядок або список) та під час кожного проходу виконує тіло циклу (малюнок 2.5).

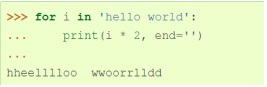


Рисунок 2.5 – Синтаксис умовного оператора For

**2.2.3 Конструкції**

Конструкція try – except використувується для обробки винятків. Винятки необхідні для виводу повідомлень про помилки програмісту.

Найпростіший приклад виключення - розподіл на нуль показаний на рисунку 2.6.

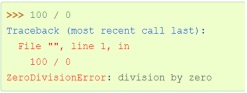


Рисунок 2.6 – Конструкція try – except

Конструкція with ... as використовується для угруповання виконання блоку інструкцій за допомогою менеджера контексту. Синтаксис конструктора with... as представлений малюнку 2.7



Рисунок 2.7 – Конструкція with... as

**2.2.4 Модулі**

Модуль у мові Python представляє окремий файл із кодом, який можна повторно використовувати в інших програмах. Для створення модуля необхідно створити власне файл із розширенням \*. py, який представлятиме модуль. Назва файлу представлятиме назву модуля. Приклад імпорту модулю в Python представлений на рисунку 2.8.

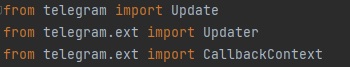


Рисунок 2.8 – приклад імпорту модулю в Python

**2.3 Принцип отримання актуальних даних з курсу криптовалют**

Оскільки курс кожної криптовалюти постійно змінюється, ми не можемо дозволити собі зберігати його значення як постійну змінну. Виходячи з цього , нам знадобиться при кожному зверненні користувача робити запит за актуальним курсом криптовалют, який можна отримати на сайтах, які спеціалізуються на цьому.

Для початку роботи чат - бота необхідно з сайту «https://www.coingecko.com» відстежувати показник потрібної нам криптовалюти, єдиний метод, який дозволяє це робити, – звернення за API.

API - опис способів, якими одна комп'ютерна програма може взаємодіяти з іншою програмою. Зазвичай входить до опису будь-якого інтернет-протоколу, програмного каркасу або стандарту викликів функцій операційної системи. Схема запиту до API представлена на рисунку 2.9.

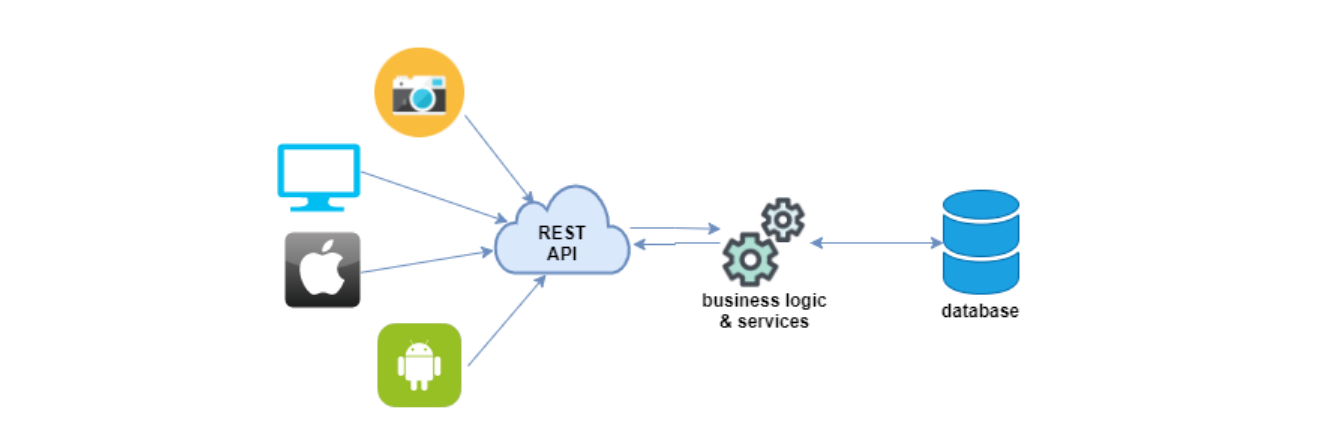


Рисунок 2.9 – Схема запиту до API

**2.4 Проектування UML діаграми варіантів використання**

Проектована система представлена у вигляді множини сутностей , які взаємодіють з системою за допомогою прецедентів. При цьому актором називається будь яка сутність, яка взаємодіє з системою. Іншими словами, кожен варіант використання визначає набір дій, які робить система при діалозі з актором.

В даній системі розглядаються такі ролі як користувач та бот.

Користувач за допомогою інтерфейсу робить необхідний йому запит до бота , який в свою чергу за допомогою інформації у запиті розуміє, що йому потрібно робити та робить запит до необхідного API для формування повідомлення з відповіддю користувачу.

На рисунку 2.10 представлена UML діаграма варіантів використання розроблюваного бота.

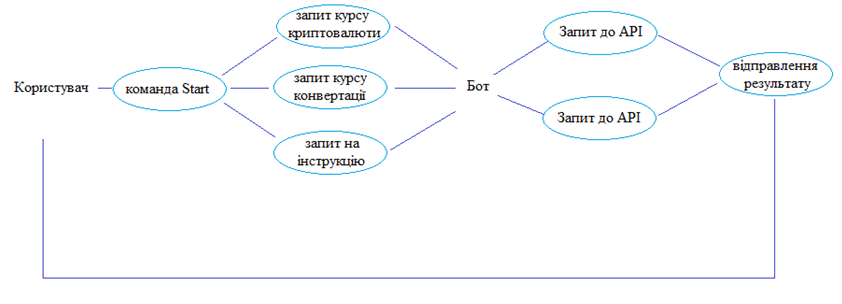


Рисунок 2.9 – UML діаграма варіантів використання

**2.5 Результати та висновки**

У цьому розділі було описано методологію розробки чат-бота, а саме розказано про головний засіб розробки чат-бота, про мову програмування Python. У цьому розділі були показані які функції, оператори та методи необхідні для написання коду для якісної роботи чат-бота. Також описано принцип отримання інформації про поточний курс криптовалют за допомогою API.

**3. ПРАКТИЧНА РЕАЛІЗАЦІЯ СЕРВІСУ ТЕЛЕГРАМ БОТА ДЛЯ**

**МОНІТОРИНГА КУРСУ КРИПТОВАЛЮТ**

**3.1 Опис процесів TO-BE**

Процеси , які будуть виконуватись проектованим ПЗ , прогнозуються шляхом створення моделі TO-BE, тобто моделей організації нових процесів ПЗ.

Використання функціональної моделі TO-BE дозволяє не лише скоротити час впровадження ІТ-системи, але й знизити ризики, пов'язані з його розробкойю. Модель TO-BE необхідна для аналізу альтернативних (найкращих) способів виконання функцій та прогнозування роботи розроблюваного ПЗ майбутньому.

Функціональна модель TO-BE чітко визначить розподіл ресурсів між операціями бізнес-процесів, що дозволить оцінити ефективність використання ресурсів після проектування.

Розглянемо зміни в результаті впровадження чат-бота, за допомогою моделі його процесів (рисунок 3.1).

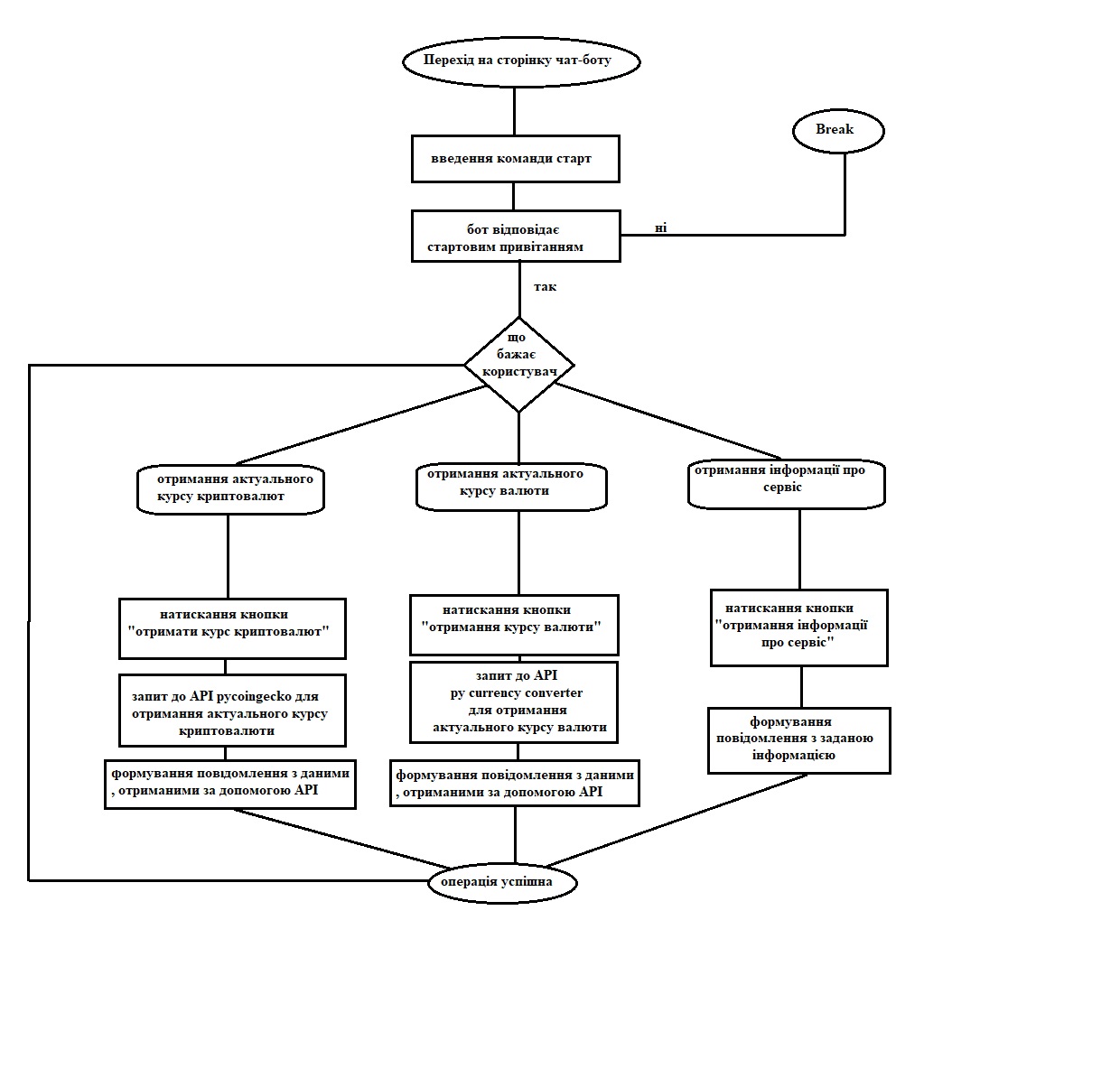


Рисунок 3.1 Модель процесів обробки звернень користувача TO-BE

Отже, модель TO-BE (рисунок 3.1) складається також із шести етапів:

* Звернення користувача;
* Команда на старт роботи бота;
* Вибір необхідного користувачу функціоналу;
* Звернення за API.
* Формування відповіді
* Завершення операції

Список учасників процесу , що розглядається у моделі TO-BE такий: клієнт, чат – бот.

Розглянемо докладніше процес обробки звернень клієнтів в чат після застосування чат-бота. Точкою входу даного процесу є звернення клієнта в чат, а точкою виходу – закриття чату. Після звернення клієнта до чату, бот очікує на команду Start для початку роботи, далі чатбот, написаний мовою python відразу після отримання команди автоматично виводить кнопки для взаємодії зі своїм функціоналом. Після того , як клієнтом було обрано функцію , наприклад запит на отримання актуального курсу криптовалют , бот робить запит за API до сервісу , який передає ці дані та формулює повідомлення – відповідь для користувача. Після виведення повідомлення бот переходить у режим очікування на наступну команду користувача.

Таким чином у результаті введення нового програмного продукту, а саме чат-бота, буде значно спрощено процес моніторингу криптовалюти та за потребою курсу валюти.

**3.2 Структура проекту**

Розробка ІТ-проекту з впровадження чат-боту для моніторингу актуального курсу криптовалют включає 6 основних етапів:

* підготовка проекту;
* аналіз;
* дизайн;
* розробка та тестування;
* розгортання;
* початковий супровід.

Структура проекту:

1) Початок роботи

2) Підготовка проекту:

- попереднє планування проекту;

- формування команди проекту;

- розробка та затвердження статуту проекту;

- визначення очікуваних результатів, термінів, кордонів та бюджету

проекту.

3) Аналіз:

- аналіз бізнес-процесів управління клієнтського сервісу;

- побудова моделі TO BE;

- аналіз технічних та програмних засобів реалізації проекту.

4) Дизайн:

- підготовка та затвердження технічного завдання;

- розробка та узгодження дизайн-рішення;

- розробка та узгодження макетів інтерфейсу.

5) Розробка та тестування:

- розробка алгоритму роботи чат-бота;

- встановлення необхідних бібліотек для Python;

- написання коду програми мовою Python;

- інтеграція з месенджером «Telegram»;

- реалізація розроблених макетів інтерфейсу;

- проведення функціонального тестування;

- усунення виявлених невідповідностей.

6) Розгортання:

- покупка сервера для розміщення ПЗ;

- розміщення ПЗ на сервері;

- складання інструкцій користувача;

- навчання персоналу.

7) Початковий супровід:

- виявлення помилок роботи програми;

- усунення помилок;

- запуск системи в експлуатацію;

8) Завершення роботи над проектом.

**3.3 Ризики проекту**

В даний час управління ризиками є одним з найбільш актуальних та динамічних напрямів стратегічного та оперативного менеджменту.

Вважається, що якісне управління ризиками дозволяє використовувати оптимальні за ефективністю та витратами кошти контролю ризиків, що відповідають поточним цілям та завданням. У таблиці 2 розглядаються ризики проекту.

Таблиця 3.1 – Ризики проекту:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| п\п | Ризик | Робота\ресурс | Стратегія запобігання  ризику | План реакції |
| Ресурсні ризики | | | | |
| 1 | Використання недосвідченого розробника | Розробник | Залучення на допомогу  досвідчених працівників;  Навчання недосвідчених  співробітників | Призначити у  допомогу  досвідченого  співробітника;  Призначити  навчання |
| 2 | Ресурси з великим обсягом робіт | Розробник | Розробка та складання  плану, за можливостями  співробітника;  погодження термінів  виконання | Залучення  додаткових  співробітників для  швидшої  реалізації |
| Календарні ризики | | | | |
| 1 | Завдання з  попередніми  тривалостями | Написання коду  програми на  Python | Залучення досвідченого  співробітника для реалізації  завдання у строк | Додати до  допомоги більш  досвідченого  співробітника |
| 2 | Надто короткі завдання | Усунення виявлених невідповідовностей | Додавання завдань, які запобігають невідповідності | Додати до плану  завдання:  «Функціональне  тестування» |

План реакції на ризик:

Для усунення ризиків усунення виявлених невідповідностей, ми додали додаткове завдання «Функціональне тестування». Цей розділ допоможе запобігти помилкам, що виникли в результаті розробки проекту, завдяки чому потрібна мінімальна кількість часу на етап "Усунення виявлених невідповідностей".

Ризики, пов'язані з ресурсами, запобігти заздалегідь можна шляхом призначення навчання ще до початку проекту. Завдяки цьому ми підвищимо рівень кваліфікованості персоналу, а також забезпечимо закриття проекту вчасно.

**3.4 Практична реалізація чат-бота мовою програмування Pyhton**

**3.4.1 Отримання токену для функціонування боту**

Для отримання токену для функціонування боту , виконаємо наступні інструкції:

1) Перейltvj до бота: @BotFather.

2) Натиснемо кнопку: START

3) Віддамо боту команду: /newbot.

4) Бот запитає, як назвати нового бота. Задамо ім’я ActualCrypto.

5) Далі потрібно ввести нік бота, щоб він закінчувався на слово bot. Введемо ActualCryptoBot.

На виході отримаємо token бота , а саме:



Рисунок 3.2 API token боту

**3.4.2 Написання програмного коду роботи бота мовою Python**

Для роботи з Telegram імпортуємо необхідні модулі :

from telegram import Update  
from telegram.ext import Updater  
from telegram.ext import CallbackContext  
from telegram.ext import Filters  
from telegram.ext import MessageHandler  
from telegram import KeyboardButton  
from telegram import ReplyKeyboardMarkup  
from telegram import ReplyKeyboardRemove

Імортуємо модуль для отримання API актуального курсу криптовалюти :

from pycoingecko import CoinGeckoAPI

Імортуємо модуль для отримання API актуального курсу конвертації валют :

from py\_currency\_converter import convert

Пропишемо логіку функціонування кнопок для вибору функції бота за допомогою функцій :

Кнопка передивлення курсу криптовалюти:

def button\_view\_handler(update: Update, context: CallbackContext):  
 price = cg.get\_price(ids='bitcoin,litecoin,ethereum', vs\_currencies='usd')  
 update.message.reply\_text(  
 text=f"Bitcoin $ {price['bitcoin']['usd']:.2f}"  
 + f"\nLitecoin $ {price['litecoin']['usd']:.2f}"  
 + f"\nEthereum $ {price['ethereum']['usd']:.2f}"  
 )

Кнопка передивлення курсу конвертації:

def button\_usd\_handler(update: Update, context: CallbackContext):  
 course = convert(amount=1, to=['RUB', 'EUR', 'UAH'])  
 update.message.reply\_text(  
 text=f"1 USD to RUB {course['RUB']}"  
 + f"\n1 USD to EUR {course['EUR']}"  
 + f"\n1 USD to UAH {course['UAH']}"  
 )

Кнопка передивлення інструкцій по роботі з ботом:

def button\_ablockchain\_handler(update: Update, context: CallbackContext):  
 update.message.reply\_text(  
 text='Привет! Для получения курса криптовалют , пожалуйста нажми кнопку "просмотреть курс криптовалют" ! Для получения актуального курса обмена валют , пожалуйста нажми кнопку "просмотреть курс валюты"'  
 )

Далі пропишемо загальну логіку роботи бота за допомогою функції та конструкції if\elif\else:

def message\_handler(update: Update, context: CallbackContext):  
 text = update.message.text  
 if text == view:  
 return button\_view\_handler(update=update, context=context)  
 elif text == usd:  
 return button\_usd\_handler(update=update, context=context)  
 elif text == ablockchain:  
 return button\_ablockchain\_handler(update=update, context=context)  
  
 reply\_markup = ReplyKeyboardMarkup(  
 keyboard=[  
 [  
 KeyboardButton(text=view),  
 KeyboardButton(text=usd),  
 KeyboardButton(text=ablockchain),  
  
 ],  
 ],  
 resize\_keyboard=True,  
 )  
 update.message.reply\_text(  
 text='Бот для получения самой актуальной информации о курсе криптовалют приветствует вас !',  
 reply\_markup=reply\_markup,  
 )

Останнім кроком задамо логіку запуску роботи бота:

def main():  
 print('Start')  
 updater = Updater(  
 token='5591804119:AAHsGKPWpPReW\_obzSV0Vl\_rDCRpL7UH4V4',  
 use\_context=True,  
 )  
 updater.dispatcher.add\_handler(MessageHandler(filters=Filters.all, callback=message\_handler))  
 updater.start\_polling()  
 updater.idle()  
if \_\_name\_\_=='\_\_main\_\_':  
 main()

**3.4.3 Результати роботи бота**

Для тестування роботи проектованого чат-боту , відкриємо його за допомогою Telegram та звернемося до усіх проектованих функцій:

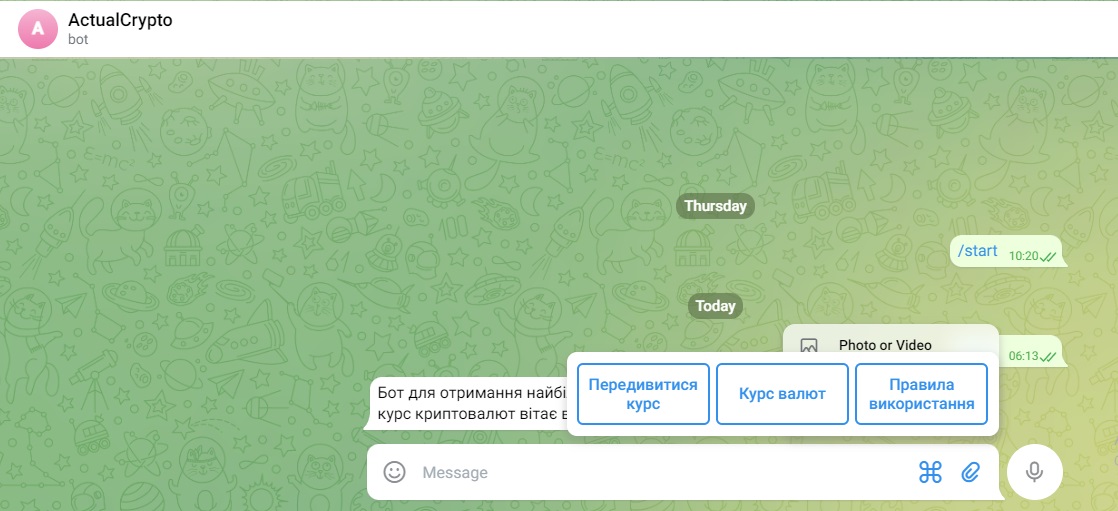


Рисунок 3.3 інтерфейс чат-боту



Рисунок 3.4 функція моніторингу актуального курсу криптовалют

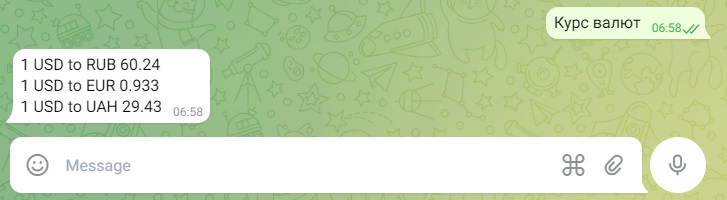


Рисунок 3.5 функція моніторингу актуального курсу конвертування валю

**ВИСНОВКИ**

Під час виконання дипломної роботи була проведена робота з вивчення технологій розробки чат-ботів з використанням мови Python. За підсумками виконання роботи було розроблено чат-бота для моніторингу курсу криптовалют.

Реалізовано функції користувача, які забезпечать можливостями перегляду актуального курсу криптовалют, курсу конвертації валют та передивлення інструкції по роботі за допомогою чат-бота у Telegram.

Розроблено діаграми ризиків та процесів для проектування та програмування системи. Після реалізації системи обраними мовою програмування та середовищем, було проведено випробування на працездатність, яке засвідчило, що система виконуємо усі поставлені нами функції.

Оскільки порушень не виявлено, всі функції відповідають вимогам, тому веб-ресурс готовий до повної функціональної роботі, тобто система для впровадження, яка може бути використана користувачем для моніторингу курсу криптовалют.

У ході роботи були виконані такі завдання:

- вивчено поняття чат-бота та його функцій;

- розглянуті переваги використання месенджерів для бізнесу;

- розглянуті мови програмування та обрано мову для розробки чат-бота;

- розробдена архітектуру програмного продукту;

- описані процеси для впровадження чат-бота;

- розроблено ІТ-проект чат-бота;

- протестовано розроблені функції програмного продукту;

**СПИСОК ЛІТЕРАТУРИ**

1. Лапшин С. Н. Архітектура підприємства [Текст]: Навчальне електронне

текстове видання / С. Н. Лапшина - ФДАОУ ВО "УрФУ", 2012. с.

2. Компанія – СКБ Контур [Електронний ресурс] – URL:

https://kontur.ru/about (дата звернення: 09.05.2020).

3. Використання MS Project для керування проектами [Електронний

ресурс]. – URL: https://habr.com/post/151593/ (дата звернення:

09.05.2020).

4. Бізнес-процеси [Електронний ресурс]. - URL: https://www.u-bs.ru/publikacii/biznes-processy.html (дата звернення: 11.05.2020).

5. Розвиток моделювання бізнес-процесів [Електронний ресурс]. -

URL: https://www.scienceforum.ru/2015/802/15636 (дата звернення:

11.05.2020).

6. AS-IS модель [Електронний ресурс]. – URL: http://pitersoft.ru/automation/more/glossary/process/as-is-model/ (дата звернення:

11.05.2020).

7. Нотація BPMN 2.0: ключові елементи та опис [Електронний

ресурс]. – URL: https://www.comindware.com/ru/blog-нотація-bpmn-2-0-

елементи-і-опис/ (дата звернення: 11.05.2020).

8. Bizagi – повнофункціональна BPM-система [Електронний ресурс]. -

URL: https://businessarchitecture.ru/bizagi/ (дата звернення: 11.05.2020).

9. Регламент обробки звернень клієнтів у чаті [Електронний

ресурс]. - URL:

https://wiki.skbkontur.ru/pages/viewpage.action?pageId=313263686 (дата

звернення: 11.05.2020).

10. Функціональна модель TO BE [Електронний ресурс]. - URL:

http://www.itstan.ru/funk-strukt-analiz/funkcionalnaja-model-to-be.html

(Дата звернення: 12.05.2020).

11. Лутц М. Вивчаємо Python, 4-те видання. - Пров. з англ. - СПб.: Символ-Плюс, 2011. - 1280 с.

12. Златопольський Д.М. Основи програмування мовою Python. - М.: ДМК Прес, 2017. - 284 с.

13. Лутц М. Програмування на Python, тому I, 4-те видання. - Пров. з англ. - СПб.: Символ-Плюс, 2011. - 992 с.

14. Лутц М. Програмування на Python, том II, 4-те видання. - Пров. з англ. - СПб.: Символ-Плюс, 2011. - 992 с.

15. Геддіс Т. Починаємо програмувати на Python. - 4-е вид.: Пер. з англ. - СПб.: БХВ-Петербург, 2019. - 768 с.

16. Лучано Рамальо Python. До вершин майстерності. - М.: ДМК Прес, 2016. - 768 с.

17. Свейгарт, Ел. Автоматизація рутини задач за допомогою Python: практичний посібник для початківців. Пров. з англ. - М.: Вільямc, 2016. - 592 с.

18. Рейтц К., Шлюссер Т. Автостопом з Python. - СПб.: Пітер, 2017. - 336 с.: іл. – (Серія "Бестселери O'Reilly").

19. Любанович Білл Простий Python. Сучасний стиль програмування. - СПб.: Пітер, 2016. - 480 с.: - (Серія "Бестсеппери O'Reilly").

20. Федоров, Д. Ю. Програмування мовою високого рівня Python: навчальний посібник для прикладного бакалаврату URL: https://urait.ru/bcode/437489

21. Шелудько, В. М. Основи програмування мовою високого рівня Python: навчальний посібник / В. М. Шелудько. URL: http://www.iprbookshop.ru/87461.html (дата звернення: 13.02.2020

22. Шелудько, В. М. Мова програмування високого рівня Python. Функції, структури даних, додаткові модулі: навчальний посібник- URL: http://www.iprbookshop.ru/87530.html

**ДОДАТКИ**

ДОДАТОК А

Програмний код

from telegram import Update  
from telegram.ext import Updater  
from telegram.ext import CallbackContext  
from telegram.ext import Filters  
from telegram.ext import MessageHandler  
from telegram import KeyboardButton  
from telegram import ReplyKeyboardMarkup  
from telegram import ReplyKeyboardRemove  
from pycoingecko import CoinGeckoAPI  
from py\_currency\_converter import convert  
  
cg = CoinGeckoAPI()  
  
view = "Посмотреть курс"  
usd = 'Курс валют'  
ablockchain = 'Прочитать'  
  
  
# кнопка для просмотра курса  
def button\_view\_handler(update: Update, context: CallbackContext):  
 price = cg.get\_price(ids='bitcoin,litecoin,ethereum', vs\_currencies='usd')  
 update.message.reply\_text(  
 text=f"Bitcoin $ {price['bitcoin']['usd']:.2f}"  
 + f"\nLitecoin $ {price['litecoin']['usd']:.2f}"  
 + f"\nEthereum $ {price['ethereum']['usd']:.2f}"  
 )  
  
# кнопка для прочтения  
def button\_ablockchain\_handler(update: Update, context: CallbackContext):  
 update.message.reply\_text(  
 text='Привет! Для получения курса криптовалют , пожалуйста нажми кнопку "просмотреть курс криптовалют" ! Для получения актуального курса обмена валют , пожалуйста нажми кнопку "просмотреть курс валюты"'  
 )  
  
  
  
# кнопка для конвертации  
def button\_usd\_handler(update: Update, context: CallbackContext):  
 course = convert(amount=1, to=['RUB', 'EUR', 'UAH'])  
 update.message.reply\_text(  
 text=f"1 USD to RUB {course['RUB']}"  
 + f"\n1 USD to EUR {course['EUR']}"  
 + f"\n1 USD to UAH {course['UAH']}"  
 )  
  
def message\_handler(update: Update, context: CallbackContext):  
 text = update.message.text  
 if text == view:  
 return button\_view\_handler(update=update, context=context)  
 elif text == usd:  
 return button\_usd\_handler(update=update, context=context)  
 elif text == ablockchain:  
 return button\_ablockchain\_handler(update=update, context=context)  
  
 reply\_markup = ReplyKeyboardMarkup(  
 keyboard=[  
 [  
 KeyboardButton(text=view),  
 KeyboardButton(text=usd),  
 KeyboardButton(text=ablockchain),  
 ],  
 ],  
 resize\_keyboard=True,  
 )  
 update.message.reply\_text(  
 text='Бот для получения самой актуальной информации о курсе криптовалют приветствует вас !',  
 reply\_markup=reply\_markup,  
 )  
  
def main():  
 print('Start')  
 updater = Updater(  
 token='5591804119:AAHsGKPWpPReW\_obzSV0Vl\_rDCRpL7UH4V4',  
 use\_context=True,  
 )  
 updater.dispatcher.add\_handler(MessageHandler(filters=Filters.all, callback=message\_handler))  
 updater.start\_polling()  
 updater.idle()  
if \_\_name\_\_=='\_\_main\_\_':  
 main()

**ДОДАТОК Б**

Скріншоти функіонування боту

