

Міністерство освіти і науки України  
Національний університет «Одеська політехніка»  
Інститут комп'ютерних систем  
Кафедра інформаційних систем

Матюхіна Олександра Олександрівна,  
Горохов Максим Денисович,  
Клейстер Руслан Сергійович  
студенти групи УП-211

ДИСЦИПЛІНА  
Об'єктно-орієнтоване програмування

КУРСОВА РОБОТА  
Розробка телеграм боту для пошуку

Спеціальність:  
122 Комп'ютерні науки

Освітня програма:  
Комп'ютерні науки

Керівник:  
Годовиченко Микола Анатолійович,  
кандидат технічних наук, доцент

Одеса – 2023

## ЗМІСТ

Анотація	3
Вступ	5
1 Огляд систем-аналогів та технологій їх розробки	6
1.1 Особливості використання телеграм ботів для ведення збору інформації	6
1.2 Огляд телеграм ботів для пошуку хобі	7
1.3 Формування вимог до основних функцій телеграм боту	8
1.4 Огляд інформаційних технологій для розробки телеграм боту	9
1.5 Висновки до першого розділу	10
2 Проектування телеграм боту для пошуку хобі	11
2.1 Мета та задачі телеграм боту	11
2.2 Визначення функціональних вимог до телеграм боту	12
2.3 Формування користувацьких історій телеграм боту	14
2.4 Ідентифікація архетипу телеграм боту	16
2.5 Проектування навігаційного графу телеграм боту	16
2.6 Проектування користувацького інтерфейсу телеграм боту	17
2.7 Висновки до другого розділу	18
3 ПРОГРАМНА РЕАЛІЗАЦІЯ телеграм боту для пошуку хобі	19
3.1 Структура серверного програмного проєкту	19
3.2 Діаграма класів телеграм боту	21
3.3 Керування вихідним кодом телеграм боту	22
3.4 Функціональне тестування розробленого телеграм боту	23
3.5 Інструкція користувача для бота пошуку хобі	23
3. Висновки до третього розділу	25
ВИСНОВКИ	27
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	28

## АНОТАЦІЯ

Курсова робота присвячена розробці телеграм боту для пошуку в рамках об'єктно-орієнтованого програмування. Основна мета проекту полягає в створенні простого, унікального, ефективного та інтуїтивно зрозумілого боту, який надає користувачам можливість шукати, підбирати та виставляти нагадування для своїх хобі.

У процесі розробки додатку будуть використані сучасні технології та практики розробки ботів на платформі node.js. Для забезпечення комунікації з користувачами буде використано месенджер Telegram, що дозволить забезпечити широку доступність та зручність використання боту.

Основна функціональна можливість боту включає пошук різноманітних хобі та надання існуючих в базі відео, фото та різноманітних статей, на основі вибраного під час проходження тесту хобі. Користувачі можуть здійснювати пошук за ключовими словами та отримувати відповідні результати. Додатково, бот буде здатний рекомендувати і підбирати контент на основі вказаних користувачем уподобань у процесі проходження веселого тесту.

Для забезпечення ефективності та швидкості роботи боту будуть використовуватися алгоритми оптимізації пошуку та обробки даних. Бот розроблен з використанням об'єктно-орієнтованого підходу, з використанням класів, об'єктів, модулів та інших концепцій ООП.

## ABSTRACT

The coursework is devoted to the development of a Telegram bot for search within the object-oriented programming framework. The main objective of the project is to create a simple, unique, efficient, and user-friendly bot that enables users to search, select, and set reminders for their hobbies.

During the application development process, modern technologies and practices for bot development on the node.js platform will be utilized. The Telegram messenger will be used for communication with users, ensuring wide accessibility and convenience in using the bot.

The main functional capability of the bot includes searching various hobbies and providing existing videos, photos, and diverse articles from the database based on the user's chosen hobby during the test. Users can perform keyword searches and receive corresponding results. Additionally, the bot will be capable of recommending and selecting content based on user preferences indicated during an enjoyable test.

To ensure efficiency and speed in the bot's operation, search and data processing optimization algorithms will be employed. The bot will be developed using an object-oriented approach, utilizing classes, objects, modules, and other OOP concepts.

## ВСТУП

У даній курсовій роботі був розроблений телеграм бот для пошуку з використанням принципів об'єктно-орієнтованого програмування. Основною метою проекту було створення простого, унікального, ефективного та інтуїтивно зрозумілого бота, який дозволяє користувачам шукати, відкривати та встановлювати нагадування для своїх хобі.

Розробка додатку включала в себе використання сучасних технологій та кращих практик розробки ботів на платформі node.js. Для комунікації з користувачами була обрана платформа Telegram, щоб забезпечити широку доступність та зручність використання бота.

Функціональні можливості бота включали пошук різноманітних ресурсів, пов'язаних з хобі, таких як статті, відео та фотографії. Користувачі могли здійснювати пошук за ключовими словами та отримувати відповідні результати. Крім того, бот надавав рекомендації та персоналізований контент на основі вподобань користувачів.

Для забезпечення ефективності та швидкості роботи були використані алгоритми оптимізації пошуку та обробки даних. Бот був розроблений з використанням об'єктно-орієнтованого підходу, з використанням класів, об'єктів, модулів та інших ключових концепцій ООП.

Проведено порівняльний аналіз існуючих аналогічних ботів для визначення переваг та обмежень розробленого бота. Крім того, були проведені випробування для оцінки швидкості та ефективності функціональності бота.

Отримані результати продемонстрували успішне досягнення поставлених цілей та сприяли розвитку об'єктно-орієнтованого програмування та розробки ботів на платформі node.js.

### 1.1 Особливості використання телеграм ботів для ведення збору інформації

Телеграм-боти грають важливу роль у покращенні особистої продуктивності та організації справ. Вони надають користувачам широкий спектр можливостей для доступу до інформації, організації завдань та спрощення повсякденних обов'язків. Завдяки таким ботам, люди можуть отримувати потрібну інформацію, встановлювати нагадування, керувати списками справ, виконувати рутинні завдання автоматично. Це дозволяє зберегти час та зосередитися на більш важливих та творчих справах. В цілому, телеграм-боти стають надійними помічниками у підвищенні продуктивності та зручності управління різними аспектами життя.

Ось деякі особливості використання телеграм-ботів для збору інформації:

1. Широкий охоплення користувачів: Telegram є популярною платформою для обміну повідомленнями з великою кількістю користувачів. Використовуючи телеграм-бота, ви можете залучити широку аудиторію та збирати інформацію від різноманітних користувачів.
2. Зручний інтерфейс: Telegram надає зручний інтерфейс для взаємодії з ботами. Користувачі можуть легко вводити інформацію, відповідати на запитання та навігувати по меню, що спрощує їх участь у зборі даних.
3. Взаємодія в реальному часі: Телеграм-боти дозволяють взаємодіяти з користувачами в реальному часі, що дозволяє одержувати дані миттєво. Користувачі можуть надавати інформацію, відповідати на запитання та спілкуватися з ботом, що дозволяє отримувати актуальні дані.
4. Можливості мультимедіа: Telegram підтримує різні формати медіа, включаючи текст, зображення, аудіо та відео. Це дозволяє користувачам надавати інформацію у різних форматах, що забезпечує розмаїтість і багатогранність наданих даних.

5. Налаштовувані форми та опитування: Ви можете створювати настроювані форми, опитування та анкети безпосередньо в інтерфейсі телеграм-бота. Правильно структуруючи запитання та варіанти відповідей, ви можете збирати конкретну інформацію і забезпечити її консистентність.
6. Автоматизоване зберігання даних: Телеграм-боти можуть бути налаштовані на автоматичне збереження зібраної інформації у базі даних або експорт її до зовнішніх систем. Це спрощує процес управління даними та дозволяє легко аналізувати та отримувати доступ до зібраних даних.
7. Безпека та конфіденційність: Telegram надає високу увагу безпеці та конфіденційності, включаючи криптографічне шифрування повідомлень. Це може надати користувачам відчуття довіри та конфіденційності під час надання інформації боту.
8. Інтеграція з іншими системами: Телеграм-боти можуть бути інтегровані з іншими системами та сервісами, що дозволяє безперебійно передавати дані або запускати дії на основі зібраної інформації. Це сприяє оптимізації робочих процесів та покращує загальний процес збору даних.

В цілому, телеграм-боти пропонують універсальну та зручну платформу для збору інформації, що дозволяє ефективно та інтерактивно збирати дані від великої кількості користувачів.

## 1.2 Огляд телеграм ботів для пошуку хобі

Наш телеграм-бот для пошуку хобі виключний на ринку і не має жодних прямих конкурентів. Наш бот пропонує унікальний та інноваційний підхід до забезпечення задоволення користувачів у їхньому пошуку та виборі хобі. Ми працюємо над тим, щоб забезпечити максимальну ефективність, зручність та якість обслуговування для наших клієнтів. Ви можете бути впевнені, що наш телеграм-бот стане незамінним помічником у вашому шляху до відкриття та насолоди новими хобі.

### 1.3 Формування вимог до основних функцій телеграм боту

Формулювання вимог до основних функцій телеграм-бота для пошуку хобі допоможе визначити, які функції мають бути реалізовані та яким чином користувачі будуть взаємодіяти з ботом. Основні функції, які можуть бути вимогами до бота, включають:

#### 1. Реєстрація користувача:

- Бот повинен мати можливість реєстрації нового користувача.
- Користувач повинен мати можливість ввести необхідну інформацію для реєстрації, наприклад, ім'я, вік, зацікавлення тощо.
- Введена інформація повинна бути збережена та використовуватись для подальшого пошуку хобі.

#### 2. Пошук хобі:

- Бот повинен мати можливість запитати користувача про його зацікавлення у хобі.
- Бот повинен мати базу даних з різними хобі та категоріями, щоб здійснювати пошук.
- Бот повинен мати логіку для вибору та пропозиції хобі, які найбільше відповідають інтересам користувача.

#### 3. Відображення результатів:

- Бот повинен мати можливість відобразити користувачеві результати пошуку хобі.
- Результати можуть бути представлені у вигляді списку або карточок з описом хобі та деталями.
- Користувач повинен мати можливість переглянути детальну інформацію про кожне хобі та обрати, яке його цікавить.

#### 4. Збереження обраного хобі:

- Бот повинен мати можливість зберегти обране хобі для користувача.
- Користувач повинен мати можливість переглянути та управляти своїми обраними хобі.

#### 5. Взаємодія з ботом:

- Бот повинен мати можливість розпізнавати команди та повідомлення користувача.
- Бот повинен мати логіку обробки команд, наприклад, команда для пошуку хобі або команда для перегляду обраних хобі.

Ці вимоги допоможуть вам розробити більш детальний план роботи над вашим телеграм-ботом і реалізувати необхідну функціональність для задоволення потреб користувачів.



## 1.4 Огляд інформаційних технологій для розробки телеграм боту

### Фреймворк Node.js

Node.js - це відкрите середовище виконання JavaScript, побудоване на движку V8 JavaScript, що розробляється Google. Воно дозволяє виконувати JavaScript на стороні сервера, забезпечуючи швидке та масштабоване створення мережеских додатків.

Node.js має наступні ключові особливості:

Асинхронний та подієвий - Node.js працює за принципом неблокуючого введення/виведення (non-blocking I/O), що дозволяє ефективно обробляти багато одночасних запитів без блокування потоку виконання. Це досягається за допомогою використання асинхронного програмування та подійного циклу.

Скальованість - Node.js розроблено з урахуванням можливості легкого масштабування додатків. Його однопотокова архітектура забезпечує високу продуктивність та можливість обробки багатьох одночасних з'єднань.

Пакетний менеджер npm - Node.js поставляється з пакетним менеджером npm, який є найбільшою колекцією безкоштовних, відкритих пакетів програмного забезпечення. npm дозволяє розробникам швидко встановлювати, оновлювати та управляти залежностями проекту.

Розширення за допомогою модулів - Node.js має модульну структуру, яка дозволяє використовувати зовнішні модулі для розширення можливостей фреймворка. Завдяки широкій спільноті розробників, в Node.js доступні тисячі модулів для різних потреб, від серверного рендерингу веб-сторінок до роботи з базами даних.

Веб-сервер - Node.js може виступати як веб-сервер, що дозволяє обробляти запити від клієнтів та повертати відповіді. Вбудована бібліотека HTTP дозволяє легко створювати серверні додатки та реалізовувати HTTP протокол.

Розробка API - Node.js добре підходить для створення власних API. Завдяки можливості обробки запитів одночасно та неблокуючій архітектурі, Node.js може швидко відповідати на запити та обробляти великий обсяг даних.

Підтримка різних ОС - Node.js доступний для різних операційних систем, включаючи Windows, macOS та різні дистрибутиви Linux, що дозволяє розробникам використовувати його на різних платформах.

Node.js відкриває широкі можливості для розробки високопродуктивних мережеских додатків, серверів, API та інших систем на основі JavaScript. Його асинхронний та подієвий підхід дозволяє створювати швидкі та масштабовані додатки, а широка спільнота та наявність пакетів допомагають прискорити процес розробки.

## 1.5 Висновки до першого розділу

У першому розділі курсової роботи проведено огляд та аналіз телеграм боту для ведення збору інформації та пошуку хобі. Було розглянуті особливості використання телеграм бота у контексті пошуку хобі. Під час аналізу були визначені функціональні вимоги до телеграм боту для пошуку хобі, зокрема, збір різноманітних даних про хобі, надання рекомендацій та варіантів хобі, а також можливість зберігання та керування списком знайдених хобі користувача.

Огляд існуючого телеграм боту для пошуку хобі дав змогу виявити їх функціонал, переваги та недоліки, що допомогло визначити, які можливості можна включити у власний телеграм бот для пошуку хобі.

Встановлення функціональних вимог є важливим кроком у процесі розробки, оскільки вони надають чітку спрямованість розробці, допомагають уникнути неоднозначностей та непорозумінь, а також забезпечують врахування потреб та вимог користувачів. Крім того, функціональні вимоги слугують основою для комунікації між учасниками проекту і дозволяють визначити обсяг роботи та потребу в ресурсах для реалізації бота.

## 2.1 Мета та задачі телеграм боту

Мета телеграм-бота для пошуку хобі полягає в наданні користувачам зручного та ефективного інструменту для відкриття та вибору нових хобі. Головною метою бота є полегшення процесу пошуку і вибору хобі, щоб користувачі могли знайти щось, що їх зацікавить та надихне.

Задачі телеграм-бота для пошуку хобі можуть включати:

1. Збір інформації про хобі: бот повинен мати базу даних з різноманітними хобі, з якої він може видавати користувачеві потрібну інформацію. Це можуть бути описи хобі, поради по початку та розвитку, матеріали для вивчення тощо.
2. Пошук хобі: користувачі повинні мати можливість шукати хобі за різними критеріями, такими як категорія, складність, часові рамки, витрати, зацікавлення та інші. Бот повинен пропонувати рекомендації, враховуючи введені користувачем критерії.
3. Розширений пошук: бот може мати функцію розширеного пошуку, яка дозволяє користувачам знаходити хобі на основі конкретних словесних запитів або фраз. Наприклад, користувач може ввести "фотографія в природі" і отримати рекомендації стосовно фотографії природи або відповідних активностей.
4. Персоналізовані рекомендації: бот може створювати профілі користувачів на основі їхніх інтересів і виборів, що допомагає надавати персоналізовані рекомендації щодо хобі. Бот може аналізувати взаємодію користувачів з попередніми рекомендаціями та враховувати їхній фідбек.

5. Доступ до інформації про хобі: бот може надавати користувачам детальну інформацію про різні хобі, включаючи опис, матеріали для вивчення, посилання на додаткові ресурси та інше.
6. Збереження хобі: бот може надавати можливість користувачам зберігати обрані хобі, створювати власні списки або збірки, додавати позначки або коментарі для подальшого зручного доступу до них.

Telegram-бот для пошуку хобі сприяє розширенню інтересів та розвитку особистості. Він допомагає знайти нові заняття, розважитися, розвивати навички та відкривати нові можливості. Завдяки боту користувачі можуть знайти хобі, які вони люблять, та займатися ними з ентузіазмом та задоволенням.

## 2.2 Визначення функціональних вимог до телеграм боту

В першому розділі курсової роботи "Telegram для пошуку хобі" було проведено огляд та аналіз телеграм ботів для ведення збору інформації. Були розглянуті особливості використання телеграм ботів у контексті пошуку хобі.

У пункті 1.1 були розглянуті особливості використання телеграм ботів для ведення збору інформації. Було з'ясовано, що телеграм боти є зручним та ефективним інструментом для збору різноманітних даних, таких як відгуки, рейтинги, рекомендації та інше.

У пункті 1.2 був проведений огляд телеграм ботів для пошуку хобі. Були розглянуті різні існуючі боти, які надають можливості для пошуку хобі та відповідних заходів. Аналіз ботів дозволив визначити їх функціонал, переваги та недоліки, а також зрозуміти, які можливості можна включити у власний телеграм бот для пошуку хобі.

У пункті 1.3 було сформовано вимоги до основних функцій телеграм боту для пошуку хобі. Вимоги були визначені на основі аналізу існуючих ботів, а також враховують потреби користувачів у пошуку різноманітних хобі.

У пункті 1.4.1 був описаний фреймворк Node.js. Вказано, що Node.js є відкритим середовищем виконання JavaScript, яке дозволяє виконувати JavaScript на стороні сервера. Були наведені основні особливості Node.js, такі як асинхронність та подієва модель, що забезпечують ефективну обробку багатьох запитів одночасно. Також було зазначено, що використання фреймворку Node.js дозволить реалізувати функціональність телеграм боту для пошуку хобі.

Загалом, перший розділ курсової роботи надає вступне розуміння теми "Телеграм для пошуку хобі" і визначає основні аспекти використання телеграм ботів для збору інформації, огляд та вимоги до таких ботів, а також виокремлює фреймворк Node.js як технологію для подальшої реалізації.

Визначення функціональних вимог до телеграм бота для пошуку хобі

Визначення функціональних вимог є ключовим кроком у процесі розробки телеграм бота для пошуку хобі. Функціональні вимоги визначають, які конкретні функції та можливості повинен мати бот. Вони надають чітку спрямованість розробки, допомагають уникнути неоднозначностей та непорозумінь.

Крім того, визначення функціональних вимог дозволяє зосередитися на потребах та вимогах користувачів. Вони допомагають врахувати, які функції та можливості будуть найбільш корисними для користувачів бота.

Також, функціональні вимоги слугують основою для комунікації між розробниками, дизайнерами та іншими учасниками проекту. Вони допомагають зрозуміти, що саме потрібно реалізувати та які очікувані результати.

Встановлення функціональних вимог дозволяє визначити обсяг роботи та потребу в ресурсах для реалізації бота. Це допомагає планувати час, бюджет та ресурси проекту ефективно.

Основним актором телеграм бота є актор "користувач". Користувач має доступ до всіх функцій бота та несе відповідальність за його використання.

З метою визначення користувацьких історій та нефункціональних вимог до телеграм бота для пошуку хобі, було розроблено діаграму сценаріїв використання бота (рис. 1.1). Діаграма сценаріїв UML (Unified Modeling Language) - це графічний інструмент для опису функціональної взаємодії між користувачами та системою.

## 2.3 Формування користувацьких історій телеграм боту

Формування сценаріїв використання телеграм-бота для пошуку хобі

Використання сценаріїв, таких як користувацькі історії, є важливою частиною розробки телеграм-бота для пошуку хобі. Ці сценарії допомагають уявити, як користувачі будуть взаємодіяти з ботом і яким чином вони зможуть знайти бажане хобі. Нижче наведено приклади користувацьких історій для телеграм-бота:

### 1. Користувач хоче знайти нове хобі:

- Користувач запускає бота в телеграмі.
- Бот привітає користувача і запропонує вибрати категорію хобі або ввести власний запит.
- Користувач обирає категорію хобі, наприклад, "музика".
- Бот надає список хобі в категорії "музика" та пропонує користувачеві переглянути детальну інформацію про кожне хобі.
- Користувач переглядає список хобі та обирає "гра на гітарі".
- Бот надає детальну інформацію про гру на гітарі, включаючи рекомендації для початківців та посилання на відеоуроки.
- Користувач вирішує спробувати гру на гітарі і зберігає це хобі в своєму профілі.

### 2. Користувач хоче знайти хобі з певними критеріями:

- Користувач запускає бота в телеграмі.
- Бот привітає користувача і запитує, які критерії хобі він шукає.

- Користувач вказує критерії, такі як "заняття на вулиці", "одноденне", "безкоштовне".
- Бот аналізує базу даних хобі і пропонує варіанти, що відповідають вказаним критеріям.
- Користувач переглядає варіанти хобі та обирає "пікнік у парку".
- Бот надає детальну інформацію про пікнік у парку, включаючи місце, дату та рекомендації щодо проведення.
- Користувач погоджується з варіантом і додає його до свого списку збережених хобі.

### 3. Користувач хоче отримати рекомендації щодо покращення поточного хобі:

- Користувач запускає бота в телеграмі.
- Бот привітає користувача і запитує, яке хобі він вже має.
- Користувач вказує своє поточне хобі, наприклад, "фотографія".
- Бот надає користувачеві інформацію про різні способи покращення навичок у фотографії.
- Користувач переглядає рекомендації та обирає "вивчення техніків обробки фотографій".
- Бот надає користувачеві посилання на онлайн-курси та ресурси для вивчення технік обробки фотографій.
- Користувач дякує за рекомендації та зберігає їх у своєму профілі.

Ці користувацькі історії використовуються для створення сценаріїв, які визначають послідовність кроків, які користувачі будуть виконувати під час взаємодії з телеграм-ботом для пошуку хобі. Залежно від потреб і можливостей бота, можуть бути розроблені й інші користувацькі історії для відображення більш широкого спектру функціональностей.

## 2.4 Ідентифікація архетипу телеграм боту

Ідентифікація архетипу телеграм боту полягає у визначенні класифікації та типу додатку на основі його функцій та характеристик. Телеграм бот відноситься до архетипу Chatbot (CB) - це програмний засіб, що забезпечує автоматизовану комунікацію з користувачами через месенджер Telegram. Цей архетип передбачає, що бот буде взаємодіяти з користувачами у режимі чату, надавати інформацію, виконувати завдання та відповідати на їх запити. Телеграм бот для пошуку хобі пропонуватиме функції пошуку, рекомендацій та інформування користувачів щодо різноманітних хобі та заходів, використовуючи інтегрований інтерфейс Telegram.

## 2.5 Проектування навігаційного графу телеграм боту

Проектування навігаційної структури телеграм бота для пошуку хобі

Навігаційна структура телеграм бота для пошуку хобі відображає зв'язки та переходи між різними екранами та функціями бота. Для проектування навігаційної структури можна використовувати вбудований функціонал Telegram Bot API, який дозволяє визначити набір команд та функцій, доступних користувачам.

Навігаційна структура бота може включати такі екрани та їх функціонал:

Головний екран: це початковий екран, на якому користувач може вибрати різні опції та функції, такі як пошук хобі, рекомендації, налаштування тощо.

Екран пошуку хобі: користувач може ввести ключові слова або параметри пошуку для знаходження певного хобі. Бот виконує пошук та надає результати.

Екран рекомендацій: на цьому екрані бот може рекомендувати користувачеві популярні та цікаві хобі, враховуючи його попередні вибори та інтереси.

Екран налаштувань: користувач може налаштувати свої вподобання та обрані категорії хобі, а також змінити інші параметри бота.

Навігаційна структура бота може мати такі переходи та зв'язки:



З головного екрана можна перейти до екрану пошуку хобі або екрану рекомендацій, вибравши відповідну опцію.

З екрану пошуку хобі можна повернутися до головного екрана або переглянути деталі певного хобі.

З екрану рекомендацій можна повернутися до головного екрана або переглянути деталі рекомендованого хобі.

З екрану налаштувань можна повернутися до головного екрана після внесення змін.

Проектування навігаційної структури допомагає зрозуміти, як користувач буде взаємодіяти з ботом, які функції та опції будуть доступні на кожному етапі та як виконуватимуться переходи між екранами.

## 2.6 Проектування користувацького інтерфейсу телеграм боту

Користувацький інтерфейс телеграм-боту для пошуку хобі може бути зручним і простим для використання. Основною метою такого інтерфейсу є допомога користувачам знайти та вибрати цікаве хобі відповідно до їхніх інтересів.

Основне вікно інтерфейсу може містити наступні елементи:

Привітальне повідомлення: Після запуску бота, користувач може бути вітаним привітальним повідомленням, що вітає його і пояснює, як використовувати бота.

Поле введення: Користувач може мати можливість ввести свої інтереси або ключові слова, пов'язані з хобі, в поле введення. Наприклад, він може ввести "фотографія" або "малювання".

Кнопка "Пошук": Після введення інтересів користувачем, він може натиснути кнопку "Пошук", щоб розпочати пошук хобі, пов'язаних з введеними інтересами.

Результати пошуку: Після натискання кнопки "Пошук", бот може відобразити список результатів пошуку. Кожен результат може мати назву хобі, короткий опис і, можливо, зображення, щоб залучити увагу користувача.

Кнопки вибору: Для кожного результату пошуку можуть бути кнопки вибору, які дозволяють користувачу позначити, чи зацікавило його конкретне хобі.

Наприклад, кнопки "Обрати" або "Не цікавить".

Повідомлення та рекомендації: Після вибору користувача, бот може надіслати повідомлення з подальшими кроками або рекомендаціями щодо хобі, які можуть бути цікаві на основі вибору користувача.

Команди та довідка: Користувач може мати доступ до додаткових команд або довідкової інформації, яка допоможе йому навігувати в інтерфейсі бота.

Наприклад, команда `/help` може відобразити доступні команди або пояснити функціональні можливості бота.

Зворотний зв'язок: Користувач може мати можливість надіслати зворотний зв'язок боту, щоб повідомити про проблеми, запитати питання або поділитися враженнями.

Важливо, щоб користувацький інтерфейс бота був зрозумілим, легким у використанні і забезпечував зручну навігацію користувача. Це може сприяти більш позитивному досвіду користувача та підвищити ефективність бота в пошуку хобі.

## 2.7 Висновки до другого розділу

У даній курсовій роботі було розглянуто різні аспекти розробки телеграм боту для пошуку хобі. Було проведено ідентифікацію архетипу телеграм боту, визначено його класифікацію та тип, який передбачає взаємодію з користувачами через месенджер Telegram. Телеграм бот для пошуку хобі буде надавати функції пошуку, рекомендацій та інформування користувачів про різноманітні хобі.

Також було розглянуто проектування навігаційного графу телеграм боту.

Навігаційна структура включає головний екран, екран пошуку хобі, екран рекомендацій та екран налаштувань. Користувач може переходити між цими екранами, залежно від своїх потреб та виборів.

Також було розглянуто проектування користувацького інтерфейсу телеграм боту. Основні елементи інтерфейсу включають привітальне повідомлення, поле введення, кнопку "Пошук", результати пошуку, кнопки вибору, повідомлення та рекомендації, команди та довідку, а також зворотний зв'язок. Користувацький інтерфейс бота повинен бути зрозумілим, простим у використанні та забезпечувати зручну навігацію користувача.

Результатом розробки телеграм боту для пошуку хобі буде зручний і ефективний інструмент, що допоможе користувачам знайти цікаві хобі відповідно до їхніх інтересів.

### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТЕЛЕГРАМ БОТУ ДЛЯ ПОШУКУ ХОБІ

#### 3.1 Структура серверного програмного проєкту

Структура серверного додатку телеграм-бота для пошуку хобі може бути організована наступним чином:

Папка "src": Основна папка проєкту, яка містить усі вихідні файли серверного додатку.

Папка "bot": Містить файл Bot.ts, який відповідає за взаємодію з Telegram API та обробку повідомлень від користувачів.

Папка "Buttons": Містить файл або файли, що визначають клавіші (кнопки) для взаємодії з користувачем у чат-боті.

Папка "CommandControllers": Містить файли, що відповідають за обробку команд, введених користувачем.

Папка "Rows": Містить файл або файли, що визначають рядки (повідомлення) для взаємодії з користувачем у чат-боті.

Файл "App.ts": Головний файл додатку, який ініціалізує та налаштовує серверний додаток.

Файл "Quiz.ts": Файл, що містить логіку опитування користувачів для визначення їхніх інтересів у хобі.

Папка "Cache": Містить файли, пов'язані з кешуванням даних для покращення продуктивності додатку.

Папка "interfaces": Містить файли, які визначають різні інтерфейси (типи даних) для використання в додатку.

Папка "module": Містить файли, які описують модулі системи, такі як модуль для роботи з хобі, категоріями, тощо.

Файл "db.ts": Файл, що містить налаштування та з'єднання з базою даних.

Файл "redis.ts": Файл, що містить налаштування та з'єднання з Redis для кешування.

Папка "schema": Містить файли, що визначають схеми бази даних, такі як схеми для хобі, категорій, користувачів тощо.

Папка "services": Містить файли, що відповідають за бізнес-логіку додатку, такі як сервіси для хобі, категорій, користувачів тощо.

Папка "static": Містить статичні файли, які можуть використовуватись у додатку, наприклад, тексти повідомлень або дані для відображення.

Папка "types": Містить файли, що визначають типи даних для TypeScript.

Файл ".env": Файл, що містить конфігураційні параметри додатку, наприклад, ключі API.

Інші файли та папки: У цій структурі можуть бути додаткові файли та папки, які використовуються для додаткових функціональних або організаційних потреб. Наприклад, папка "util" може містити утилітарні функції для використання в додатку, а папка "decorators" може містити декоратори для розширення функціональності.

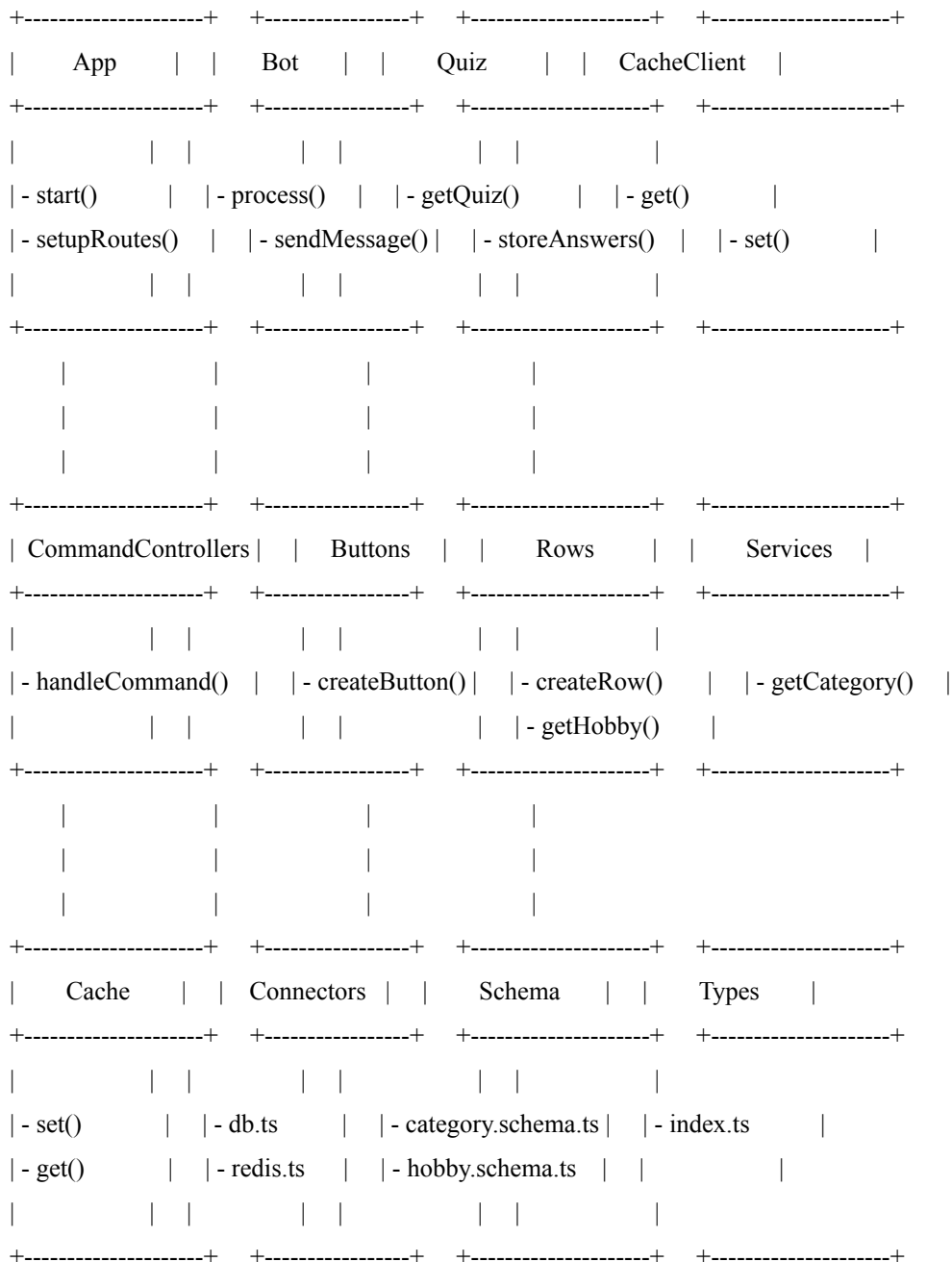
Ця структура серверного додатку дозволяє організовувати код на основі функціональності та модульності, зручно використовувати різні сервіси, контролери та схеми даних, а також забезпечує чистоту та структурованість проекту.

### 3.2 Діаграма класів веб-ресурсу

Ось діаграма класів для попередньої структури:

scss

Copy code



Ця діаграма класів відображає основні класи та їх взаємозв'язки. Клас App відповідає за запуск та налаштування серверного додатку. Клас Bot взаємодіє з Telegram API та обробляє повідомлення від користувачів. Клас Quiz відповідає за проведення опитування користувачів. Клас CacheClient взаємодіє з кешем для зберігання та отримання даних.

Класи `CommandControllers`, `Buttons`, `Rows` та `Services` представляють різні контролери та рядки для взаємодії з користувачем. У класі `CommandControllers` знаходяться методи для обробки команд користувача. Клас `Buttons` містить методи для створення кнопок. Клас `Rows` відповідає за створення рядків повідомлень. Клас `Services` містить різні сервіси для обробки даних, такі як категорії та хобі. Класи `Cache`, `Connectors`, `Schema` та `Types` представляють компоненти, пов'язані з кешуванням, з'єднанням з базою даних, схемами даних та типами. Клас `Cache` має методи для роботи з кешем. Клас `Connectors` містить з'єднання з базою даних. Клас `Schema` визначає схеми даних. Клас `Types` містить визначення типів даних. Ця діаграма допомагає зрозуміти взаємозв'язки між класами та їхню взаємодію в структурі додатку.

### 3.3 Керування вихідним кодом телеграм боту

Використання системи контролю версій є дуже важливим для будь-якого програмного проекту, в тому числі і для веб-ресурсу продуктової крамниці. Використання системи контролю версій дозволяє зберігати історію всіх змін, які були внесені до веб-ресурсу.

Це дозволяє в разі потреби повернутися до попередніх версій веб-ресурсу та відновити попередній стан, який був на момент попередньої версії.

Також, система контролю версій дозволяє контролювати версії веб-ресурсу, що значно полегшує роботу з веб-ресурсом та дозволяє підтримувати його в актуальному стані.

Веб-ресурс- Серверна частина Кількість комітів - 8 Кількість pull-реквестів - 0  
Кількість гілок у репозиторії - 1

### 3.4 Функціональне тестування розробленого телеграм боту

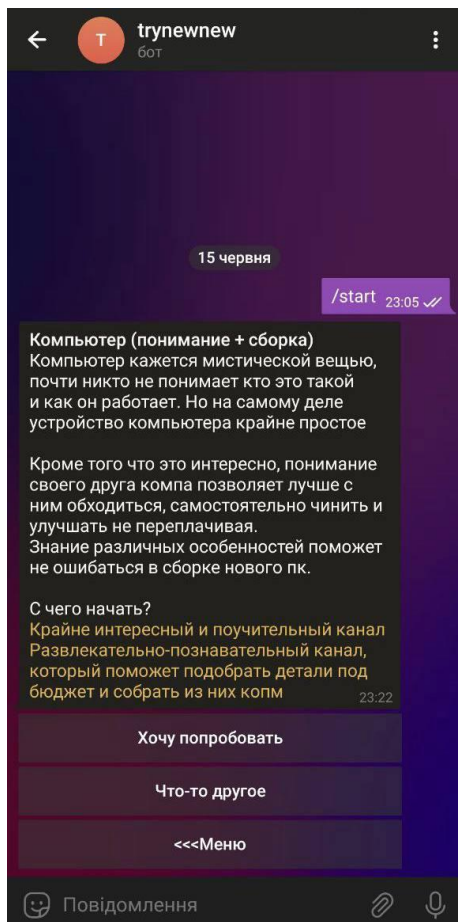


Рисунок 3.1

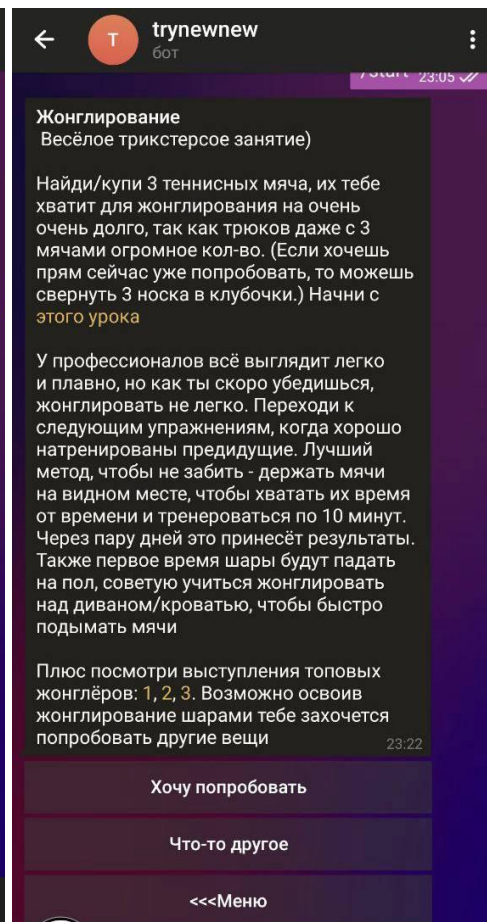


Рисунок 3.2

### 3.5 Інструкція користувача для бота пошуку хобі

Вітаємо вас з використанням бота пошуку хобі! Цей бот допоможе вам знайти цікаві заняття та хобі, які вам можуть сподобатися.

Для початку, натисніть кнопку "Start" або надішліть команду /start, щоб розпочати взаємодію з ботом.

Після запуску бота, ви будете отримувати повідомлення від бота зі списком хобі і запитаннями, що допоможуть нам краще зрозуміти ваші інтереси.

Щоб переглянути категорії хобі, натисніть кнопку "Переглянути категорії" або надішліть команду /categories. Ви побачите список доступних категорій, таких як спорт, музика, мистецтво, гра, подорожі та багато інших.

Виберіть категорію, яка вас цікавить, натиснувши відповідну кнопку або надіславши назву категорії у повідомленні. Наприклад, якщо вас цікавить спорт, натисніть кнопку "Спорт" або надішліть повідомлення з текстом "Спорт".

Після вибору категорії, ви будете отримувати пропозиції хобі, пов'язані з цією категорією. Ви можете переглянути деталі кожного хобі, натиснувши на нього або відправивши команду `/hobby followed_by_hobby_name`. Наприклад, `/hobby Футбол`.

Якщо ви не задоволені пропозиціями хобі або хочете отримати випадкове хобі, натисніть кнопку "Випадкове хобі" або надішліть команду `/randomhobby`.

Ви також можете відповісти на запитання бота, щоб уточнити ваші інтереси.

Наприклад, якщо бот запитує про ваші уподобання в музиці, ви можете відповісти, наприклад, "Рок" або "Електронна музика".

У будь-який момент ви можете надіслати команду `/help`, щоб отримати список доступних команд та інформацію про взаємодію з ботом.

Якщо у вас виникли труднощі або у вас є питання, ви можете звернутися до команди підтримки, надіславши повідомлення адміністратору або натиснувши кнопку "Контакти".

Насолоджуйтесь пошуком нових хобі та забавляйтесь з ботом!

Зауваження: Бот пошуку хобі є частиною курсової роботи і може містити обмежену кількість хобі та функціональності.



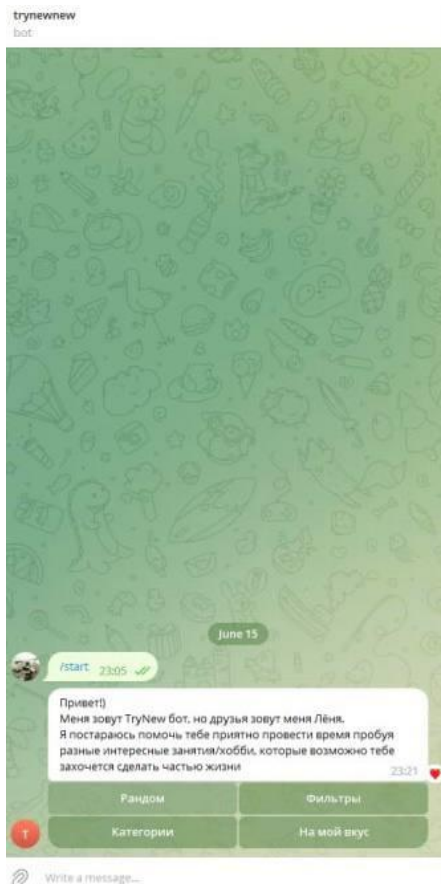


Рисунок 3.3

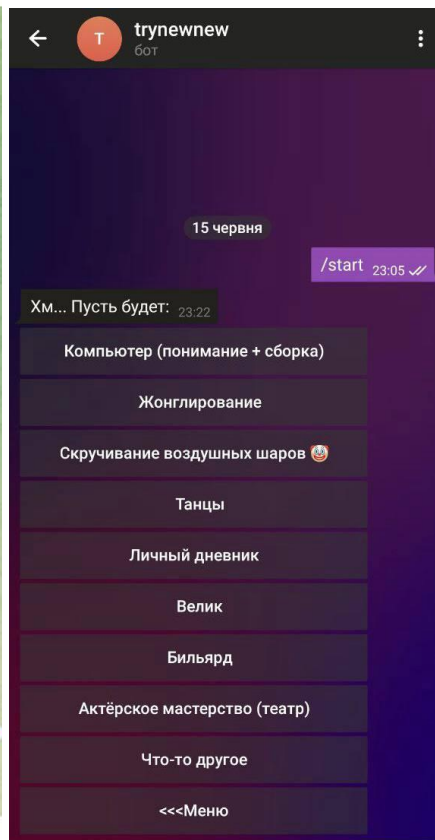


Рисунок 3.4

### Висновки до третього розділу

Структура серверного додатку телеграм-бота для пошуку хобі включає різні папки і файли, організовані згідно функціональності та модульності. Головна папка проекту містить основні файли серверного додатку, такі як Bot.ts для взаємодії з Telegram API та обробки повідомлень, App.ts для ініціалізації та налаштування додатку, а також Quiz.ts для логіки опитування користувачів.

Додаткові папки включають Buttons для визначення клавiш (кнопок), CommandControllers для обробки команд користувачів, Rows для взаємодії з користувачем за допомогою рядків повідомлень, Cache для кешування даних, interfaces для визначення різних типів даних, module для опису модулів системи,

таких як хобі та категорії, а також schema для визначення схем бази даних і services для бізнес-логіки додатку.

Діаграма класів відображає основні класи та їх взаємозв'язки. Клас App відповідає за запуск та налаштування серверного додатку, Bot взаємодіє з Telegram API, Quiz проводить опитування користувачів, а CacheClient забезпечує роботу з кешем. Класи CommandControllers, Buttons, Rows та Services виконують різні функції для взаємодії з користувачем, а класи Cache, Connectors, Schema та Types представляють компоненти, пов'язані з кешуванням, з'єднанням з базою даних, схемами та типами даних.

Ця структура дозволяє зручно організовувати код, забезпечує чистоту та структурованість проекту, а також полегшує розширення функціональності додатку. Діаграма класів допомагає зрозуміти взаємозв'язки між класами та їхню взаємодію в структурі додатку.

## ВИСНОВКИ

У даній курсовій роботі був розроблений телеграм бот для пошуку з використанням принципів об'єктно-орієнтованого програмування. Основною метою проекту було створення простого, унікального, ефективного та інтуїтивно зрозумілого бота, який дозволяє користувачам шукати, відкривати та встановлювати нагадування для своїх хобі.

Розробка додатку включала в себе використання сучасних технологій та кращих практик розробки ботів на платформі node.js. Для комунікації з користувачами була обрана платформа Telegram, щоб забезпечити широку доступність та зручність використання бота.

Функціональні можливості бота включали пошук різноманітних ресурсів, пов'язаних з хобі, таких як статті, відео та фотографії. Користувачі могли здійснювати пошук за ключовими словами та отримувати відповідні результати. Крім того, бот надавав рекомендації та персоналізований контент на основі вподобань користувачів.

Для забезпечення ефективності та швидкості роботи були використані алгоритми оптимізації пошуку та обробки даних. Бот був розроблений з використанням об'єктно-орієнтованого підходу, з використанням класів, об'єктів, модулів та інших ключових концепцій ООП.

Проведено порівняльний аналіз існуючих аналогічних ботів для визначення переваг та обмежень розробленого бота. Крім того, були проведені випробування для оцінки швидкості та ефективності функціональності бота.

Отримані результати продемонстрували успішне досягнення поставлених цілей та сприяли розвитку об'єктно-орієнтованого програмування та розробки ботів на платформі node.js.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Node.js – an open-source, cross-platform JavaScript runtime environment. URL: <https://nodejs.org/en/docs>
2. Node.js Telegram Bot API  
URL: <https://www.npmjs.com/package/node-telegram-bot-api>
3. Stack overflow  
URL: <https://stackoverflow.com>
4. Node-telegram-bot-api-tutorial  
URL: <https://github.com/hosein2398/node-telegram-bot-api-tutorial>
5. Telegram APIs documentation  
URL: <https://core.telegram.org/>
6. URL: <https://www.freecodecamp.org/news/solid-principles-explained-in-plain-english/>
7. Node.js – an open-source, cross-platform JavaScript runtime environment. URL: <https://nodejs.org/en>
8. Express – fast, unopinionated, minimalist web framework for Node.js.  
URL: <https://expressjs.com/>
9. MySQL Home.  
URL: <https://www.mysql.com/>
10. PostgreSQL:  
URL: <https://www.postgresql.org/>
11. MongoDB Atlas Database  
URL: <https://bit.ly/414ADjX>