

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем

**Кваліфікаційна робота
на здобуття ступеня бакалавра**

за спеціальністю 121 Інженерія програмного забезпечення
на тему:

**TELEGRAM-БОТ ДЛЯ ДИСТАНЦІЙНОГО КУРСУ ДИСКРЕТНОЇ
МАТЕМАТИКИ**

Виконав студент 4-го курсу
Микола СИЗЬКО

(підпис)

Науковий керівник:
доцент, кандидат фізико-математичних наук
Володимир ШЕВЧЕНКО

(підпис)

Засвідчую, що в цій роботі немає запозичень
з праць інших авторів без відповідних
посилань.

Студент

(підпис)

Роботу розглянуто й допущено до захисту на
засіданні кафедри інтелектуальних
програмних систем

« ____ » _____ 2021 р.,
протокол № ____

Завідувач кафедри
Олександр ПРОВОТАР

(підпис)

Київ-2021

РЕФЕРАТ

Обсяг роботи 41 сторінок, 13 ілюстрацій, 2 таблиці, 10 джерел посилань.

КЛЮЧОВІ СЛОВА: БОТ, ТЕЛЕГРАМ, АРІ, РНР, ЧАТ, ТЕЛЕГРАМ-БОТ, ЧАТ-БОТ, МЕСЕНДЖЕР, ДИСТАНЦІЙНЕ НАВЧАННЯ, ДИСКРЕТНА МАТЕМАТИКА.

Об'єктом дослідження є процес отримання студентами навчальних матеріалів та опанування дисципліни «Дискретна математика». Предметом дослідження є програмний засіб Телеграм-бот для дистанційного курсу дискретної математики.

Метою кваліфікаційної роботи є дослідження системи Телеграм, аналіз можливостей Телеграм-ботів, створення Телеграм боту для дистанційного курсу Дискретна математика.

Методи розроблення: об'єктно-орієнтоване програмування, розробка програмного продукту. Інструменти розроблення: мова програмування РНР, текстовий редактор Sublime Text 3, Telegram Bot API, формат обміну даними JSON, програмний засіб для передача даних сURL.

Результати роботи: було досліджено різні платформи для створення Телеграм-бота та обрано мову програмування РНР. Створено бота і наповнено його лекційними та практичними матеріалами з дискретної математики. Запущено бот на сервері Факультету комп'ютерних наук та кібернетики. Отже, студенти можуть у зручній для них формі отримувати навчальні матеріали з дисципліни «Дискретна математика».

За методами розробки та інструментальними засобами робота виконувалася сумісно з роботами по створенню чат-ботів на платформі Телеграм, вивченням ООП, системного програмування та операційних систем, дискретної математики.

Програмний продукт можна використовувати для отримання навчальних матеріалів з дискретної математики, надсилання повідомлень зворотного зв'язку, вивчення дискретної математики під час дистанційного навчання.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ	4
ВСТУП	5
РОЗДІЛ 1. ЧАТ-БОТИ ТЕЛЕГРАМ.....	7
РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ СИСТЕМИ.....	14
РОЗДІЛ 3. ВИБІР ЗАСОБІВ ТА ТЕХНОЛОГІЙ РЕАЛІЗАЦІЇ	23
РОЗДІЛ 4. РЕАЛІЗАЦІЯ ПРОЕКТУ	29
ВИСНОВКИ.....	39
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	41

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

cURL - Client URL.

API - Application Programming Interface, інтерфейс прикладного програмування.

ООП – об'єктно-орієнтоване програмування.

UML - Unified Modeling Language, уніфікована мова програмування, використовується в парадигмі об'єктно-орієнтованого програмування.

SysML - Systems Modeling Language, мова моделювання загального призначення для застосувань в системній інженерії.

PHP - Hypertext Preprocessor, скриптова мова програмування.

HTML – HyperText Markup Language, мова тегів.

JSON - JavaScript Object Notation, текстовий формат обміну даними між комп'ютерами.

ВСТУП

Оцінка сучасного стану об'єкта дослідження або розробки. Завданням роботи є створення Телеграм-бота для покращення якості вивчення дистанційного курсу з дискретної математики в Київському національному університеті імені Тараса Шевченка. На поточний момент університеті успішно використовується бот для вивчення курсу Комп'ютерні мережі. Користь від використання подібних ботів полягає у підвищенні зручності та швидкості навчання. Таким чином, студенти мають змогу отримувати навчальні матеріали, завдання для самостійного опрацювання, модульні контрольні роботи. На даний момент у курсі Дискретна математика не використовуються Телеграм-боти.

Актуальність роботи та підстави для її виконання. У 2020 році через пандемію навчальні заклади перейшли на дистанційну або змішану форму навчання. Для багатьох закладів освіти це новий спосіб викладання. Підставою для створення бота для дистанційного курсу Дискретної математики є потреба в спрощенні організаційних дій викладача та студентів, зменшенні часу для отримання відповідей на запитання. Телеграм-бот – це інструмент, з використанням якого можна вирішити описані проблеми.

Мета й завдання роботи. Метою кваліфікаційної роботи є створення Телеграм боту для дистанційного курсу Дискретна математика. Було поставлено такі завдання:

- дослідити систему Телеграм;
- проаналізувати можливості Телеграм ботів;
- визначити основні функції бота для дистанційного навчання;
- створити і протестувати програмний засіб для курсу Дискретна математика.

Об'єкт, методи й засоби розроблення. Об'єктом розробки є Телеграм бот зі зручним інтерфейсом для опанування дисципліни дискретна математика. Інструментом створення програмного засобу було обрано мову програмування

PHP, інструмент передачі даних cURL та текстовий редактор Sublime Text 3, Telegram Bot API.

Взаємозв'язок з іншими роботами. За методами розробки та інструментальними засобами робота виконувалася сумісно з вивченням ООП, системного програмування та операційних систем, дискретної математики.

РОЗДІЛ 1. ЧАТ-БОТИ ТЕЛЕГРАМ

Telegram – [1] кросплатформений, безкоштовний засіб миттєвого обміну повідомленнями. Ця система має великий функціонал. З допомогою Telegram можна створювати чат-боти.

Чат-бот – це програма, призначена для обміну повідомленнями з користувачем у месенджері. З допомогою неї можна здійснювати комунікацію з великою кількістю осіб. Чат-боти самі надсилають повідомлення залежно від запиту користувача, тому їх масово використовують у різних сферах життя. Обмін повідомленнями відбувається у системі «людина – комп'ютер».

Великий функціонал ботів полягає в таких можливостях:

- 1) відправка найбільш актуальної інформації. Відразу після публікації бот може надсилати інформацію користувачеві;
- 2) інтеграція з багатьма іншими системами, наприклад з ботами;
- 3) проведення платежів від користувачів Telegram;
- 4) утиліти та інструменти. Бот може перекладати тексти, попереджувати про події;
- 5) гра з одним та багатьма користувачами. Бота можна навчити, наприклад, грати в шахи з користувачем, проводити вікторини;
- 6) соціальні сервіси. Бот може знайти співрозмовника для обговорення спільних інтересів.

Як працюють чат-боти. Telegram-бот – це особливий акаунт в системі Telegram. Для створення бота не потрібно номера телефону, на відміну від звичайних акаунтів користувачів. По суті, боти відіграють роль інтерфейсу для доступу до сервісу, що працює на віддаленому сервері.

Telegram-бот відрізняється від звичайного акаунта. Відмінності описані в таблиці 1.1.

Таблиця 1.1 – Різниця між акаунтом користувача і бота

Звичайний акаунт	Акаунт бота
Має статус активності: онлайн, був у мережі [дата або час], заходив недавно, був цього місяця в Telegram, був дуже давно.	Замість статусу відображається слово «бот».
Повідомлення зберігаються, поки їх не видалить учасник чату.	Для повідомлень ботів виділене обмежене сховище на серверах, повідомлення зникають з часом.
Може почати спілкування з користувачем Telegram.	Почати спілкування з ботом може лише користувач. Щоб знайти бота, можна використати пошук по назві бота або за посиланням виду t.me/<bot_username>.
Може присвоїти собі будь-яке ім'я користувача.	Ім'я бота повинне закінчуватись на «bot».
За замовчуванням може бачити «давні» повідомлення в груповому чаті.	За замовчуванням, при додаванні бота в груповий чат може бачити лише ті повідомлення, що надіслані після його приєднання.
Може бути не онлайн.	Працює цілодобово.

Переваги Telegram-ботів у порівнянні з мобільними додатками, соціальними мережами, сайтами:

- 1) зручність у використанні, простота. Для роботи з мобільними додатками, групами в соціальних мережах, сайтами потрібно витратити певний час на знайомство з інтерфейсом. Телеграм-боти створюються так, що робота з ними та їх інтерфейс інтуїтивно зрозумілі. Треба лише знайти потрібний для своєї задачі бот;

- 2) ефективне споживання трафіку. Інтерфейс у ботів текстовий. Якщо у користувача низька швидкість інтернет-з'єднання або обмежена кількість трафіку, то з Телеграм-ботом буде так само зручно працювати, як і при інших умовах. Звичайно, для завантаження файлів знадобиться той самий час, що й з інших ресурсів, це обумовлено розміром даних, що передаються між ботом і користувачем;
- 3) низька вартість. Телеграм бот можна запустити на персональному комп'ютері, не обов'язково витратити кошти на оренду серверів;
- 4) швидкість зв'язку. Боти миттєво відповідають на користувацькі повідомлення, якщо вони на це запрограмовані;
- 5) не потребує авторизації та встановлення. Для користування ботом треба лише перейти до діалогу з ним, якщо ви користуєтесь месенджером Телеграм. В іншому випадку, можна завантажити Телеграм на пристрій або скористатися веб версією додатку;
- 6) надійність у порівнянні з людиною. Телеграм-бот моментально відповідає на запит та надає інформацію. Не пропускає повідомлень від жодного користувача, як це може зробити людина;
- 7) швидкодія при виконанні шаблонних завдань. Боти виконують таку роботу швидко та без помилок.

Систематизація чат-ботів Телеграм.

За сферою застосування вони поділяються на:

- 1) боти для особистого спілкування;
- 2) споживчі, для підтримки клієнтів певної компанії.

Певні види ботів залежно від задач – таблиця 1.2.

Таблиця 1.2 – Види Телеграм-ботів

Боти для компаній сфери послуг	Записи, бронювання; оплата; отримання даних про компанію; зворотній зв'язок; програми лояльності; відправка повідомлень.
Боти для фінансових компаній	Отримання даних про розташування; найближчих відділень, банкоматів; інформація про банк, умови обслуговування; відповіді на часті питання; надсилання заявок користувачем; зворотній зв'язок; поповнення рахунку, погашення заборгованості; сповіщення про новини.
Боти для служб доставки їжі	Отримання інформації про заклади, їх меню; оформлення замовлення, оплата; надсилання відгуків; надсилання рекламної інформації.
Боти для служб підтримки	Відповіді на часті запитання; зв'язок з оператором; подання заявок; відслідковування статусу обробки заявок.
Боти для освітніх установ	Отримання інформації про графік освітнього процесу; матеріали лекцій; джерела інформації для додаткового опрацювання;

	матеріали з контрольними роботами; отримання оцінок; сповіщення про проведення занять; отримання відеоматеріалів.
Розважальні	Вибір фільму за певними тегами; гра з ботом в парні ігри; ігри в групових чатах з великою кількістю учасників.

Таблицю 1.2 можна продовжувати, оскільки Телеграм-ботів можна використовувати у різних сферах. Їх можливості залежать від конкретних задач та уяви розробника.

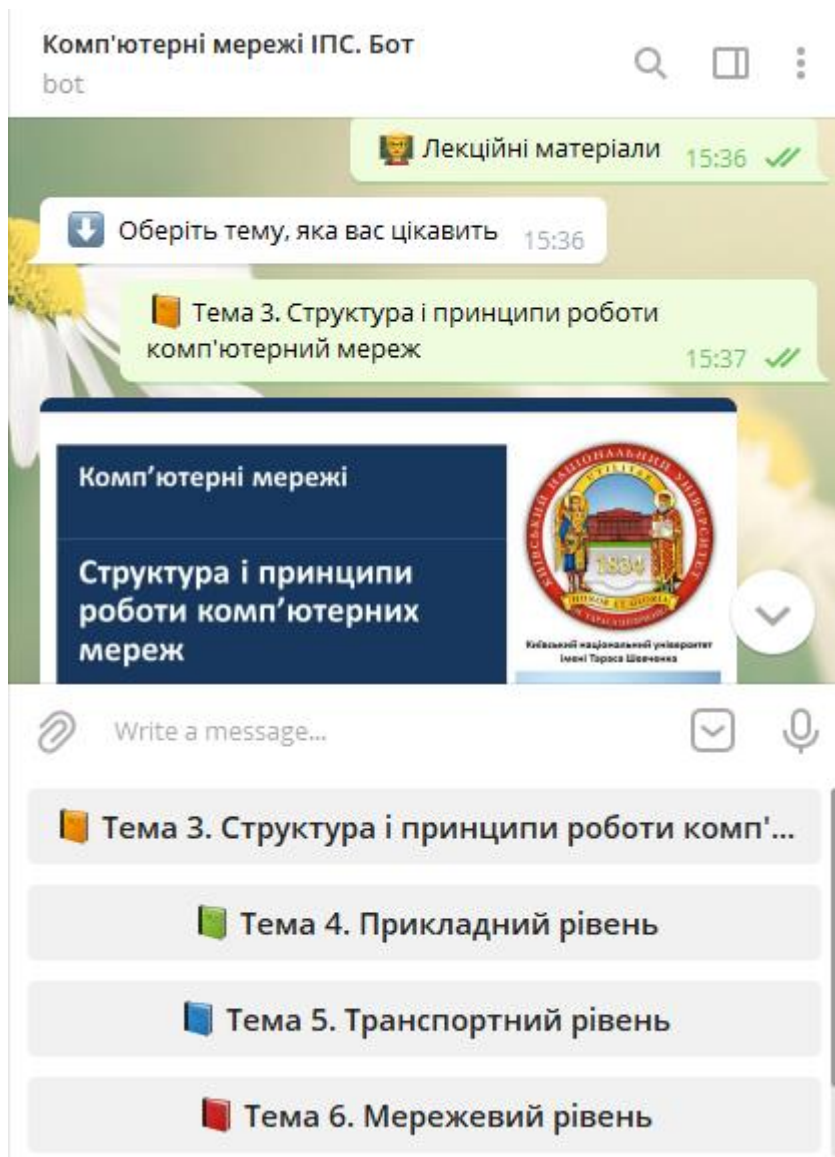
Бот можна створити не лише в месенджері Телеграм. Інші популярні платформи для створення ботів:

- 1) Facebook Messenger;
- 2) Viber;
- 3) WhatsApp;

Приклади Телеграм-ботів:

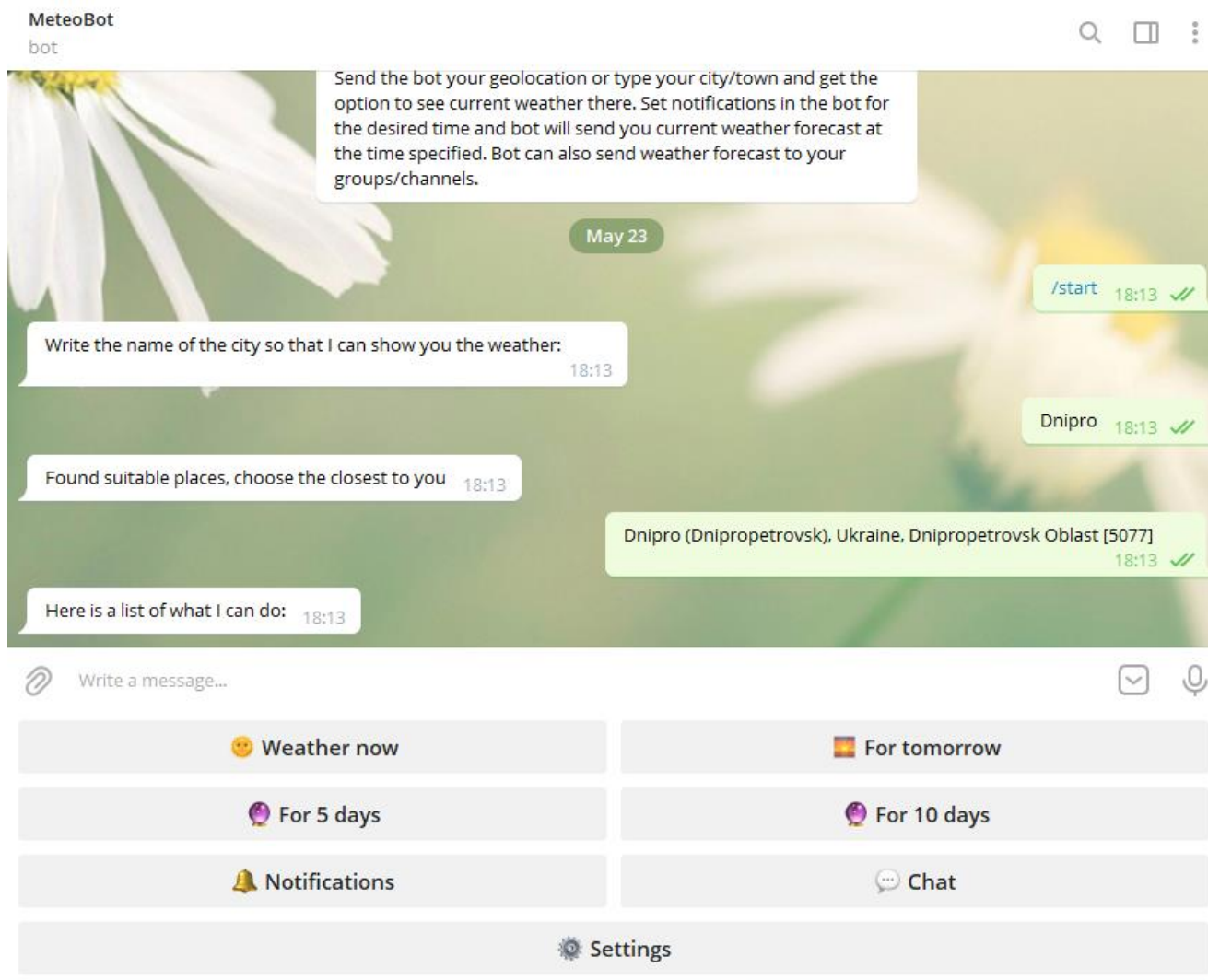
Комп'ютерні мережі ІІС. Бот (@CN_Support_Bot) - Бот інформаційної підтримки курсу «Комп'ютерні мережі», для груп ІІС-41, -42, Факультет Комп'ютерних наук та кібернетики, Київський національний університет імені Тараса Шевченка. З допомогою цього бота можна отримати лекційні матеріали, матеріали по лабораторним роботам, тестові контрольні роботи, журнал відвідування та успішності, літературу курсу «Комп'ютерні мережі». Використання бота спростило проведення організаційних заходів, пришвидшило процедуру отримання матеріалів студентами, збільшило продуктивність використання часу для навчання. Інтерфейс бота показано на рис. 1.1.

Рисунок 1.1 – Інтерфейс чат-бота «Комп'ютерні мережі ІПС. Бот»



MeteoBot (@WeathersBot) - Weather from Gismeteo.ru. Бот надсилає дані про погоду в будь-якій точці світу. Дані про прогноз погоди отримує з сайту Gismeteo.ru. Інтерфейс показано на рис. 1.2.

Рисунок 1.2 – інтерфейс чат-бота «MeteoBot»



РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ СИСТЕМИ

Аналіз і специфікація вимог до інформаційної підсистеми.

Користувачі – люди, які тим чи іншим чином відносяться до навчального процесу, впливають на нього, є виконавцями або користувачами системи.

Користувачі:

- 1) викладачі;
- 2) студенти.

Цілі Користувачів:

- 1) оптимальний зв'язок між студентами та викладачами;
- 2) зручність та швидкість використання.

Вимога — це опис того, що має бути реалізовано.

Бізнес-вимоги – цілі власника бізнесу до розробки системи. Добре розуміючи бізнес клієнта або потреба в розробці системи стає простіше реалізувати проект.

Бізнес-вимоги:

- 1) Оперативне отримання завдань онлайн у месенджері Телеграм’;
- 2) можливість відслідковувати поточні бали завдяки журналу успішності;
- 3) можливість ставити питання викладачеві онлайн.

Рисунок 2.1 - Бізнес-вимоги



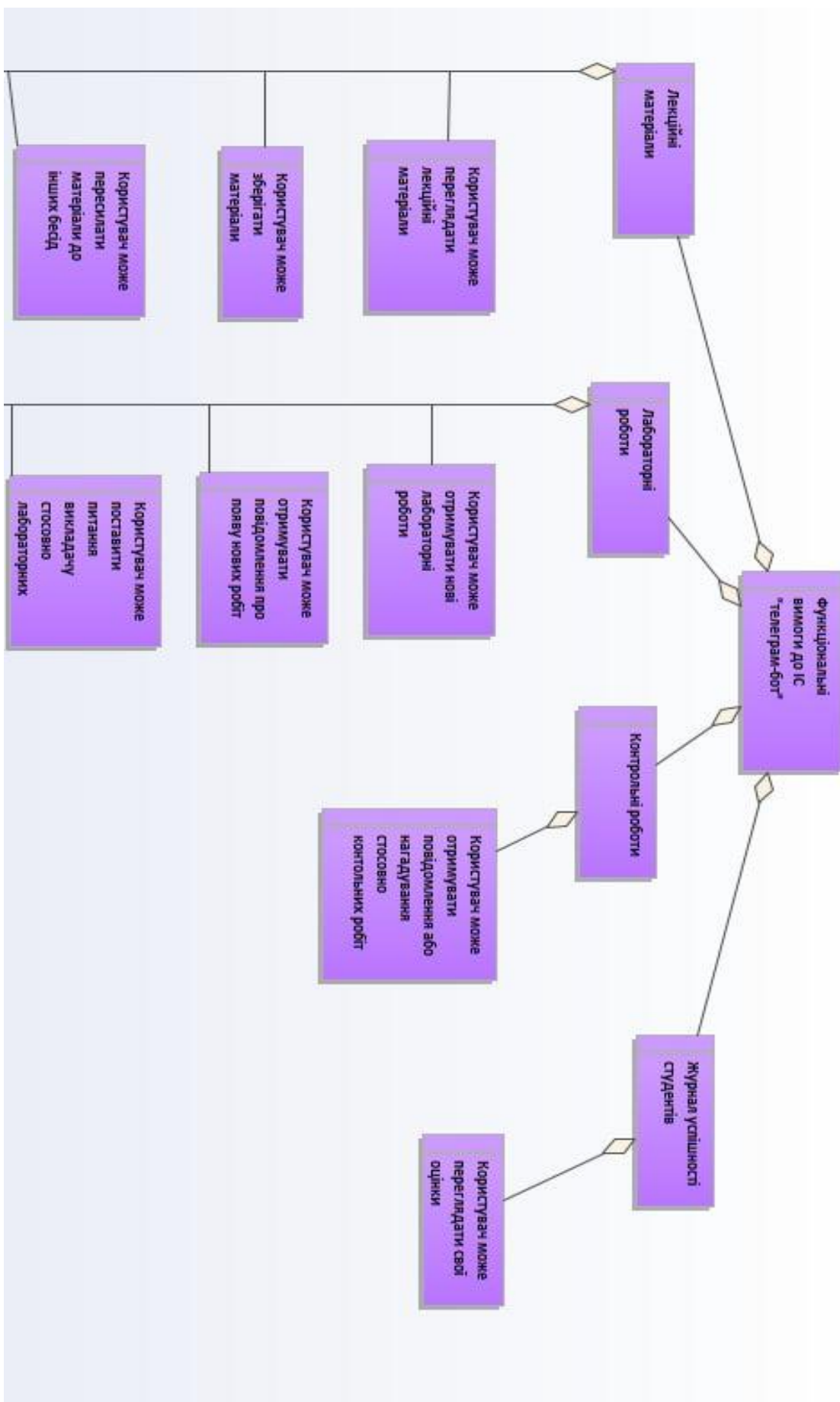
Функціональні та нефункціональні вимоги до системи. Бот, призначений для користування студентами, містить такі блоки:

- 1) блок для отримання навчальних матеріалів і практичних завдань;
- 2) вікно допомоги/підтримки;
- 3) блок відгуків.

Інтерфейс адміністратора дозволяє отримувати повідомлення від студентів, відповідати на них.

Функціональні вимоги, або функціонал системи – відповідають на питання «що робить» або які функції виконуються в програмному забезпеченні. Також функціональні вимоги дають змогу спроектувати поведінку та внутрішню роботу системи. З функціональними вимогами можна ознайомитись на рис. 2.2.

Рисунок 2.2 - Функціональні вимоги



Окрім функціональних вимог також потрібно сформулювати нефункціональні вимоги для повноти представлення усіх можливостей системи. Тож, якщо функціональні вимоги відповідають на питання «що система робить», то нефункціональні показують, якою вона повинна бути.

Нефункціональні вимоги – вимоги до системи, які задають можливість оцінювання якості її роботи – вказано на рис. 2.3.

Рисунок 2.3 - Нефункціональні вимоги



Специфікація нефункціональних вимог – перелік нефункціональних вимог з урахуванням пріоритетності вимоги та рівня складності. З рисунку 2.4 можемо побачити, що ефективність та масштабування є пріоритетною вимогою. У цьому кейсі можемо сказати, що ефективність програмного забезпечення наймовірно важлива, тому що безрезультатний додаток не має сенсу розробляти. Масштабованість - означає, що мережа дозволяє збільшити кількість вузлів і довжину з'єднання в широкому діапазоні, не знижуючи продуктивність мережі.

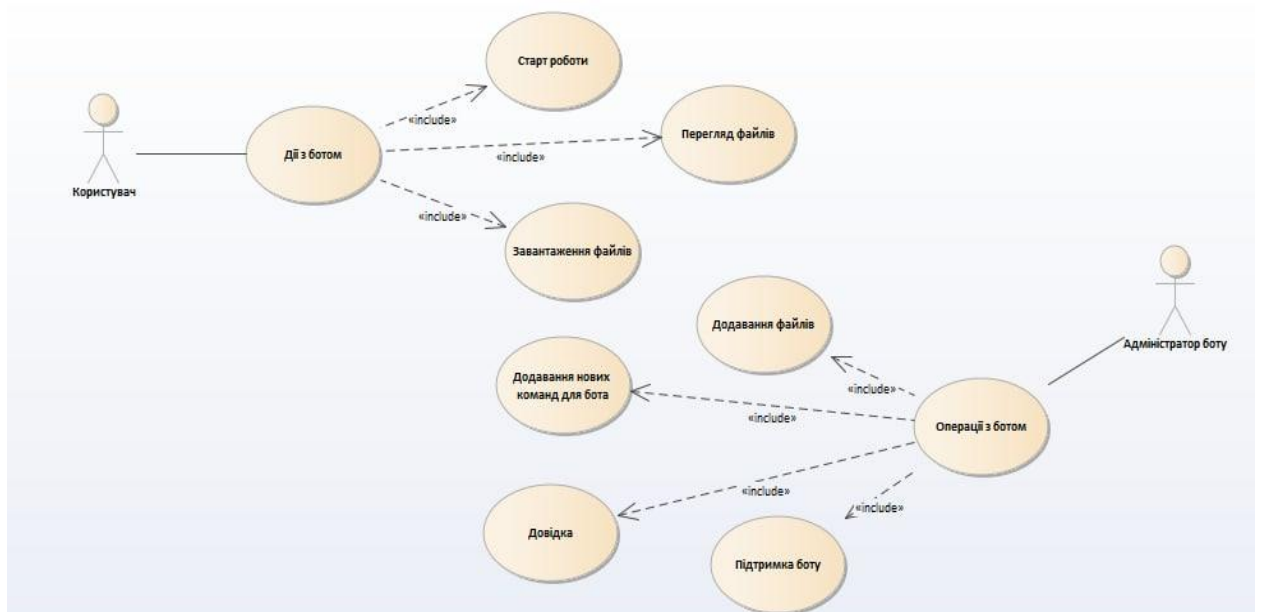
Рисунок 2.4 - Специфікація нефункціональних вимог

Item	Status	Difficulty	Priority
Відновлювальність	Proposed	Medium	Medium
Відповідність	Proposed	Medium	Medium
Доступність	Proposed	Medium	Medium
Ефективність	Proposed	Medium	High
Керування помилками	Proposed	High	Medium
Маштабування	Proposed	Medium	High
Надійність	Proposed	<u>High</u>	Medium
Підтримка	Proposed	Medium	Medium

Моделювання поведінки системи. Для моделювання поведінки системи було використано мови UML/SysML.

Діаграма прецедентів – діаграма у мові UML, що зображує відношення між акторами та прецедентами в системі. Також може інтерпретуватись як діаграма варіантів використання. Зображена на рис. 2.5.

Рисунок 2.5 - Діаграма прецедентів



Користувач боту може здійснювати такі дії:

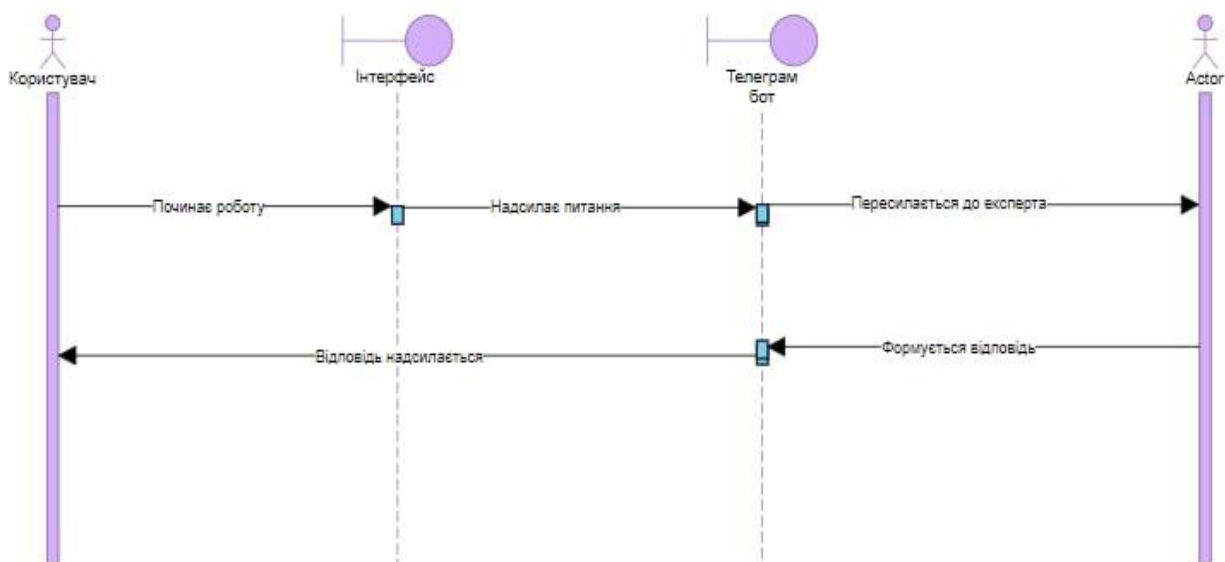
- 1) завантажувати матеріали;
- 2) переглядати матеріали;
- 3) залишати відгуки;
- 4) ставити питання викладачеві.

Дії адміністратора боту:

- 1) додавання файлів;
- 2) додавання нових команд для боту;
- 3) технічна підтримка боту.

Діаграма послідовностей (рис. 2.6) – показує взаємодію об'єктів, відсортованих за часом. Зокрема, на цих малюнках показано задіяні об'єкти та послідовність відправлених повідомлень.

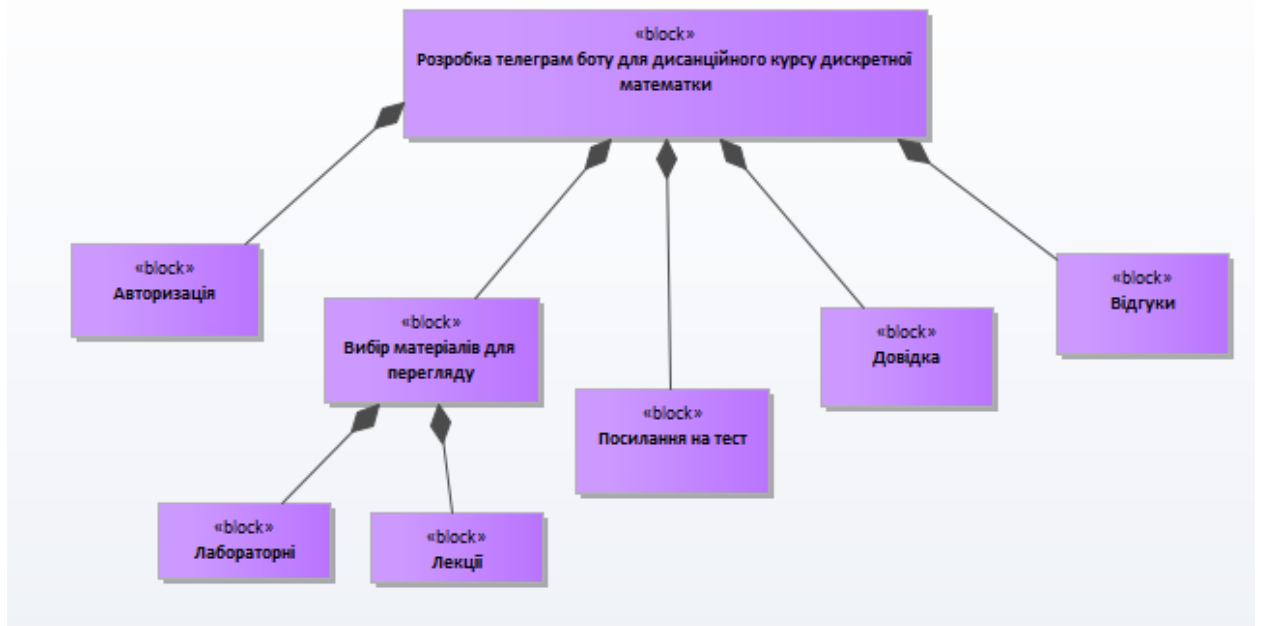
Рисунок 2.6 - Діаграма послідовностей для користувача та адміністратора системи



Моделювання структури системи.

Діаграма блоків (Block definition diagram) – це статична блок-схема, яка показує компоненти системи, їх зміст (атрибути, поведінку, обмеження), інтерфейси та взаємозв'язки. Дана діаграма проедставлена на рис. 2.7.

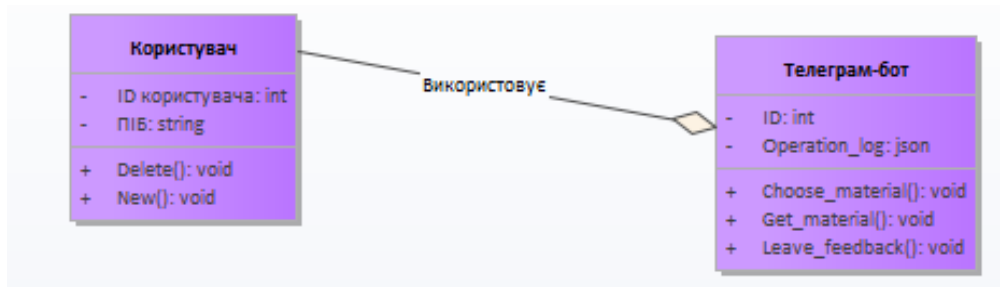
Рисунок 2.7 – Діаграма блоків



Діаграми класів (рис. 2.7) є основою будь-якої проектної документації програмного забезпечення і відправна точка для реалізації програмних додатків. Використовуючи схеми класів, ви можете формалізувати логічну модель програми на рівні структурного елемента (тобто класу) системи. Він забезпечує статичне уявлення про структуру програми - які класи вона містить, які взаємозв'язки між цими класами і що містить кожен клас.

Модель кожного класу є шаблоном для майбутніх об'єктів, які створюються на основі цього класу. Кожна програма може містити багато різних об'єктів. Згідно з концепцією ООП, відомі об'єкти мають стани та поведінку. На рівні класу стан описується полями даних, а поведінка - методами.

Рисунок 2.7 - Діаграма класів



Моделювання користувацького інтерфейсу. Інтерфейс користувача включає 4 блоки:

- 1) вікно для отримання матеріалів;
- 2) вікно для переходу на тестування;
- 3) вікно допомоги;
- 4) вікно для залишення відгуку.

Заходячи на сторінку бота, можемо натиснути кнопку «Почати». Це є командою для бота, що потрібно почати свою роботу. Такий тригер запускає ланцюг наступних повідомлень.

Заходячи на сторінку бота, можемо натиснути кнопку «Почати». Це є командою для бота, що потрібно почати свою роботу. Такий тригер запускає ланцюг наступних повідомлень. Прототип інтерфейсу показаний на рис 2.8.

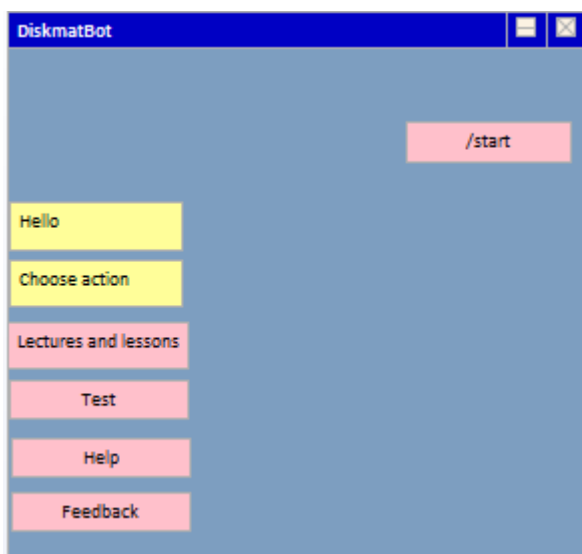
Рисунок 2.8 – Прототип користувацького інтерфейсу



Як результат, ми бачимо, що натиснувши «Почати» з'являється меню з наступними кроками. Прототип цього меню показано на рис. 2.9. Окрім цього є привітання та повідомлення з текстом «оберіть дію». Таким чином ми можемо обрати дію з доступних кнопок:

- 1) «Lectures and lessons» - означає, що натиснувши можемо обрати матеріали для підготовки;
- 2) «Test» - містить посилання на самостійну або модульну роботу;
- 3) «Help» - містить блок з інструкцією щодо використання бота;
- 4) «Feedback»- містить можливість поставити питання викладачеві або залишити відгук про роботу бота.

Рисунок 2.9 – Прототип меню Телеграм-бота



РОЗДІЛ 3. ВИБІР ЗАСОБІВ ТА ТЕХНОЛОГІЙ РЕАЛІЗАЦІЇ

Щоб створити Телеграм-бота, можна використовувати будь-яку мову програмування, з допомогою якої можна працювати з Telegram Bot API. Також існують конструктори ботів, вони потребують плату або розміщують свою рекламу в ботах. Конструктор – це шаблон, використовуючи який можна створити бота.

Розробка на JavaScript. JavaScript – [2] це мова програмування, що інтерпретується, об'єктно-орієнтована, з функціями першого класу. Використовується найбільше для веб-сторінок. Існують також більш вдосконалені версії JavaScript на стороні сервера, такі як Node.js, що дозволяють додавати веб-сайту більше функціональних можливостей, ніж завантаження файлів (наприклад, співпраця в режимі реального часу між кількома комп'ютерами). Усередині хост-середовища (наприклад, веб-браузера) JavaScript може бути підключений до об'єктів свого середовища для забезпечення програмного контролю над ними.

Програмний код на JavaScript називають скриптами. Скрипти можуть входити до HTML і виконувати свій скрипт під час завантаження сайту.

Клієнтська частина JavaScript розширює основну мову, надаючи об'єкти для управління браузером та його об'єктною моделлю документа. Наприклад, розширення на стороні клієнта дозволяють програмі розміщувати елементи у формі HTML та реагувати на дії користувача, такі як клацання миші, введення форми та навігація сторінкою.

Серверна частина JavaScript розширює основну мову, надаючи об'єкти, що мають відношення до роботи JavaScript на сервері. Наприклад, розширення на стороні сервера дозволяють програмі взаємодіяти з базою даних, забезпечують безперервність інформації від одного виклику до іншого додатка або виконують маніпуляції з файлами на сервері.

JavaScript надає розробнику такі можливості:

- 1) зміна сторінок браузера;
- 2) додавання і видалення тегів;

- 3) зміна стилів сторінок;
- 4) отримання звітів про дії користувача на сторінці;
- 5) робота з соокіе-файлами.

Перевагами цієї мови є підтримка більшості браузерів та легка взаємодія з додатками.

Телеграм-бот можна написати на мові програмування JavaScript, основними інструментами є `node-telegram-bot-API` і `Telgraf.js`.

`Telegraf` – [3] це бібліотека, що спрощує розробку власних ботів Telegram за допомогою JavaScript або TypeScript . Особливості:

- 1) Повна підтримка API Telegram Bot 5.1;
- 2) Відмінний друк TypeScript;
- 3) Займає мало місця;
- 4) Firebase / Glitch / Heroku / AWS λ / ...;
- 5) `http/https/fastify/Connect.js/express.js` сумісні веб-хуки;
- 6) Розширювана.

Приклад бота [4]:

```
import { Telegraf } from 'telegraf'

const bot = new Telegraf(process.env.BOT_TOKEN)

bot.command('quit', (ctx) => {
  // Explicit usage
  ctx.telegram.leaveChat(ctx.message.chat.id)

  // Using context shortcut
  ctx.leaveChat()
})

bot.on('text', (ctx) => {
  // Explicit usage
  ctx.telegram.sendMessage(ctx.message.chat.id, `Hello ${ctx.state.role}`)

  // Using context shortcut
  ctx.reply(`Hello ${ctx.state.role}`)
})

bot.on('callback_query', (ctx) => {
  // Explicit usage
  ctx.telegram.answerCbQuery(ctx.callbackQuery.id)

  // Using context shortcut
  ctx.answerCbQuery()
```

```

})

bot.on('inline_query', (ctx) => {
  const result = []
  // Explicit usage
  ctx.telegram.answerInlineQuery(ctx.inlineQuery.id, result)

  // Using context shortcut
  ctx.answerInlineQuery(result)
})

bot.launch()

// Enable graceful stop
process.once('SIGINT', () => bot.stop('SIGINT'))
process.once('SIGTERM', () => bot.stop('SIGTERM'))

```

Наведемо корисні механізми для розробника:

- 1) Middlewares – коли користувач надсилає повідомлення боту, то повідомлення спочатку проходить проміжну обробку, після чого надходить в контролер;
- 2) Sessions – сесії не пов’язані з контролером, можуть зберігати в собі інформацію;
- 3) Webhooks - бот може працювати тільки в двох режимах - long polling або Webhooks. Працюють вони з однаковою швидкістю, проте рекомендується використовувати Webhooks. Бот приймає сам усі оновлення, не треба надсилати запити на сервер Телеграм;
- 4) Markup – клас, що відповідає за створення кнопок. Важливий для розробки ігрових ботів.

Django – веб-фреймворк високого рівня мови програмування Python. Був розроблений для роботи з Apache і з використанням системи управління базами даних PostgreSQL. З часом Django вдосконалили і він зміг працювати з іншими продуктами управління базами даних:

- 1) MySQL;
- 2) SQLite;
- 3) Microsoft SQL Server;
- 4) DB2;
- 5) Firebird;

- 6) SQL Anywhere;
- 7) Oracle.

Django має власний ORM - (англ. Object-Relational Mapping, Об'єктно-реляційне відображення, або перетворення) – це технологія програмування, що пов'язує БД з концепціями ООП мов програмування, створює віртуальну об'єктну БД.

Можливості Django:

- 1) зв'язок з БД, підтримка запитів;
- 2) панель адміністрування;
- 3) шаблони функцій;
- 4) кеш;
- 5) з'єднання з модулями аутентифікації;
- 6) автоматично створювана документація;
- 7) вбудовані бібліотеки.

Переваги Django. Масштабованість – працює при високому трафіку. Багатогранність – завдяки великому набору шаблонів його можна використовувати для створення різноманітних сайтів та багатьох задач. Швидкість – розробник економить час на пошук рішення задачі. Безпечність – Django пропонує протестовані рішення проблем.

Недоліком є те, що попри велику кількість шаблонів їх не можна компанувати.

Система управління БД MongoDB.

MongoDB – [5] документоорієнтована СУБД, не потребує опису схеми таблиць, класифікована як NoSQL:

- 1) зберігає дані у гнучких, схожих на JSON документах, поля можуть відрізнятися від документа до документа, структура даних може змінюватися з часом;
- 2) модель документа відображається на об'єктах у кодї програми, це полегшує роботу з даними;

- 3) спеціальні запити, агрегування та індексація в режимі реального часу забезпечують потужні способи доступу та аналізу даних;
- 4) горизонтальне масштабування та географічний розподіл вбудовані, їх просто використовувати.

Розробка на Python. Для створення Телеграм-бота використовують бібліотеки та засоби зв'язку з ботами PyTelegramBotAPI та Python-telegram-bot.

Функції PyTelegramBotAPI:

- 1) обробка повідомлень;
- 2) обробка запитів зворотного зв'язку;
- 3) обробка проміжного ПЗ;
- 4) розмітка відповідей;
- 5) TeleBot;
- 6) Inline Mod;
- 7) надсилання великих текстових повідомлень;
- 8) асинхронна оставка повідомлень;
- 9) контроль кількості потоків;
- 10) використання вебхуків;
- 11) ведення журналів;
- 12) механізм слухача;
- 13) підтримка проксі.

Розробка на PHP. PHP (Hypertext Preprocessor – [6] гіпертекстовий препроцесор) – скриптова мова програмування сценаріїв загального призначення з відкритим кодом. Розроблена для виконання веб-сценаріїв на стороні сервера. Синтаксис мови походить з мов C, Java и Perl. Перевагами PHP є:

- 1) простота. Синтаксис гнучкий та інтуїтивно зрозумілий;
- 2) спільнота. PHP використовується багатьма розробниками, тому при виникненні проблем можна швидко знайти їх вирішення в мережі Інтернет;
- 3) документація. На офіційному сайті маємо можливість знайти опис усіх можливостей мови програмування;

- 4) гнучкість. PHP – це вбудована кросплатформена мова програмування. Її можна запустити на більшості платформ, що існують, це робить її гнучкою для розробки;
- 5) масштабованість - означає, що продуктивність системи використовується максимально ефективно та при збільшенні продуктивності ефективність зростає. Сайт може витримати набагато більші навантаження, за умови, якщо його розмістити на кількох серверах.

WebHook - це механізм за яким виконується зв'язка бота з Телеграмом. Коли користувач вводить команду, її треба якимось чином передавати на сервер і отримувати відповідь. Webhook створений саме для цих цілей. Він повідомляє Телеграму куди передавати команди.

JSON – [7] це текстовий формат запису даних. Він дозволяє в текстовому вигляді представити як окреме число або рядок, так і складні структури, наприклад, масиви з даними. Використання цього формату запису зручно тим, що він читабельний і інтуїтивно зрозумілий, в той же час дозволяє зберігати дуже складні структури даних. Крім того, він більш компактний, ніж xml, тому кращий для обміну даними між веб-браузером і сервером.

У мові php починаючи з версії 5.2. є такі 4 функції для роботи з json:

- 1) `json_decode` - декодує рядок JSON (з рядка json-формату отримує дані);
- 2) `json_encode` - повертає JSON-уявлення даних (перетворює дані в json-рядок);
- 3) `json_last_error_msg` - повертає рядок з повідомленням про помилку останнього виклику `json_encode ()` або `json_decode ()`;
- 4) `json_last_error` - повертає останню помилку.

РОЗДІЛ 4. РЕАЛІЗАЦІЯ ПРОЕКТУ

Для реалізації дипломного проекту було використано мову програмування PHP, cURL, JSON.

Спочатку було здійснено реєстрацію Телеграм-бота у @BotFather. Реєстрація здійснюється за 5 кроків:

- 1) З допомогою пошуку в Телеграм відкриваємо чат з @BotFather;
- 2) Вводимо або обираємо зі списку команду /newbot;
- 3) Пишемо назву DiskmatBot створюваного бота, надсилаємо повідомлення з назвою;
- 4) Пишемо юзернейм @Diskmat_Vot бота, по якому його можна буде знайти в пошуку. В кінці юзернейма обов'язково має бути слово «bot»;
- 5) За бажанням можна встановити аватар бота, опис, список команд.

Після реєстрації отримуємо токен. Токен використовується для доступу до HTTP API, інформація про токен конфіденційна.

В офіційній документації вказано, що API Telegram базується на простих запитах HTTP. Існує два різні способи обробки запиту, який користувач надішле:

- 1) перевіряти "вручну", використовуючи «Long Polling»;
- 2) довірити все Telegram і поставити "Webhook". У цьому випадку будь-який запит від користувача Telegram надсилатиме на наш на сервер.

Ми зосередимося на другому варіанті, але у нього є обмеження: ви повинні встановити на веб-сайті сертифікат SSL, щоб усі запити могли бути доставлені через захищений протокол HTTPS. Також застосовуються самопідписані та безкоштовні сертифікати Let's Encrypt, що підтримуються більшістю хостів.

Для того, щоб почати створювати програмний код для бота, треба завантажити і встановити відповідні інструменти розробки. Для роботи використано ноутбук Acer Aspire Timeline Ultra M3 з 64-розрядною версією операційно системи Windows 10. Процес підготовки засобів розробки:

- 1) завантажено php-8.0.6 з офіційного сайту <https://www.php.net/downloads.php>;
- 2) створено файл конфігурації php.ini – наповнення файлу скопійовано з php.ini-development;
- 3) розкоментовано рядок extension=curl цього файлу для використання в роботі cURL;
- 4) завантажено сертифікат Mozilla CA у форматі PEM cacert.pem з сайту <https://curl.se/docs/caextract.html>;
- 5) у файлі конфігурації php.ini вказано шлях до сертифіката: «curl.cainfo = C:\php-8.0.6-Win32-vs16-x64\extras\ssl\cacert.pem»;
- 6) завантажено та встановлено текстовий редактор Sublime Text для зручного написання програмного коду.

Розглянемо фрагмент програмного коду [10].

```
<?php
```

```
require_once 'ValueLib.php';
```

```
/*
```

Клас, що реалізує мінімальний функціонал для взаємодії з Telegram A.

```
*/
```

```
abstract class TelegramBotCore {
```

```
    /*
```

Використовують для підключення до одного з серверів Телеграму.

```
    */
```

```
    protected $host;
```

```
    protected $port;
```

```
    protected $APIUrl;
```

```
    /*
```

Інформація, за якою сервери Телеграму дізнаються інформацію про бота.

```
*/  
  
public $botId;  
public $botUsername;  
protected $botToken;  
  
//екземпляр cURL  
protected $handle;  
//створений для того, щоб не виконувати ініціалізацію більше одного разу.  
protected $inited = false;  
  
/*  
Затримки перед повторним запитом у випадку помилки.  
*/  
protected $lpDelay = 1;  
protected $netDelay = 1;  
  
/*  
Параметри подій, отримуваних під час long polling.  
*/  
protected $updatesOffset = false;  
protected $updatesLimit = 30;  
protected $updatesTimeout = 10;  
  
/*  
Обмеження на час одного запиту.  
*/  
protected $netTimeout = 10;  
protected $netConnectTimeout = 5;  
  
/*
```

Конструктор, заповнює поля класу переданими опціями.

```
*/
public function __construct($options = array()) {
    $options += array(
        'host' => 'API.telegram.org',
        'port' => 443,
    );

    $this->host = $host = $options['host'];
    $this->port = $port = $options['port'];

    $proto_part = ($port == 443 ? 'https' : 'http');
    $port_part = ($port == 443 || $port == 80) ? '' : ".$port";

    $this->APIUrl = "{$proto_part}://{host}{$port_part}/bot{$this->botToken}";
}

```

/*

Отримує з серверів Телеграма інформацію про бота.

Використовує метод getMe з офіційного Telegram API.

Встановлює флаг inited, щоб не виконувати завантаження більше, ніж один

раз.

*/

```
public function init() {
    if ($this->inited) {
        return true;
    }

    $this->handle = curl_init();
}

```

```

$response = $this->request('getMe');

if ($response === null || !$response['ok']) {
    echo "Can't connect to Telegram API\n";
    echo "Response:\n";
    var_dump($response);
    exit;
}

```

```

$bot = $response['result'];
$this->botId = $bot['id'];
$this->botUsername = $bot['username'];

```

```

$this->inited = true;
return true;
}

```

```

/*

```

Публічний метод для початку роботи методом long polling.

```

*/

```

```

public function runLongpoll() {
    $this->init();
    $this->longpoll();
}

```

```

/*

```

Використання цього та наступного методу потребує публічного IP і доменного імені, тому в даній роботі надається перевага використанню методу runLongpoll.

```

*/

```

```
public function setWebhook($url) {
    $this->init();
    $result = $this->request('setWebhook', array('url' => $url));
    return $result['ok'];
}
```

```
public function removeWebhook() {
    $this->init();
    $result = $this->request('setWebhook', array('url' => ""));
    return $result['ok'];
}
```

```
/*
```

Метод для виконання довільного запиту до серверів Телеграму з
можливістю гнучкого налаштування.

\$method - назва методу Telegram API, \$params - параметри Telegram API,
\$options - параметри cURL.

```
*/
```

```
public function request($method, $params = array(), $options = array()) {
    $options += array(
        'timeout' => $this->netTimeout,
    );
}
```

```
/*
```

файли виділяються в окремий масив, для того, щоб передавати їх у тілі
запиту. А інші параметри в URL.

```
*/
```

```
$getParams = [];
$postParams = [];
foreach ($params as $key => $value) {
    if (is_numeric($value) || is_string($value)) {
```

```

    $getParams[] = urlencode($key).'='.urlencode($value);
} else if (is_object($value) && ($value instanceof CURLFile)) {
    $postParams[$key] = $value;
} else {
    $getParams[] = urlencode($key).'='.urlencode(json_encode($value));
}
}
$queryString = implode('&', $getParams);
$url = $this->APIUrl.'/'.$method.(($queryString ? '?'.$queryString : ''));

if (count($postParams) > 0) {
    curl_setopt($this->handle, CURLOPT_POST, true);
    curl_setopt($this->handle, CURLOPT_POSTFIELDS, $postParams);
} else {

    curl_setopt($this->handle, CURLOPT_HTTPGET, true);
}

$connect_timeout = $this->netConnectTimeout;
$timeout = $options['timeout'] ?: $this->netTimeout;

curl_setopt($this->handle, CURLOPT_URL, $url);
curl_setopt($this->handle, CURLOPT_RETURNTRANSFER, true);
curl_setopt($this->handle, CURLOPT_CONNECTTIMEOUT,
$connect_timeout);
curl_setopt($this->handle, CURLOPT_TIMEOUT, $timeout);

$response_str = curl_exec($this->handle);
$errorno = curl_errno($this->handle);
if ($errorno === 60) {

```

/* Якщо не вдалося перевірити сертифікат, то пишемо про це користувачу, щоб було зрозуміло, що бракує файлу cacert.pem

```
*/
```

```
echo (curl_error($this->handle)."\n");
```

```
exit;
```

```
}
```

```
$http_code = intval(curl_getinfo($this->handle, CURLINFO_HTTP_CODE));
```

```
/*
```

Якщо в файлі конфігурації вказано неіснуючий токен, то повідомляємо про це користувачеві.

```
*/
```

```
if ($http_code == 401) {
```

```
    throw new Exception('Invalid access token provided');
```

```
} else if ($http_code >= 500 || $errno) {
```

```
    sleep($this->netDelay);
```

```
    if ($this->netDelay < 30) {
```

```
        $this->netDelay *= 2;
```

```
    }
```

```
}
```

```
$response = json_decode($response_str, true);
```

```
return $response;
```

```
}
```

```
/*
```

Здійснює опитування серверів Телеграма на наявність подій.

Для кожної події він запускає метод onUpdateReceived.

Викликає себе рекурсивно, щоб виконання бота не зупинялося після всіх оброблених подій та він чекав на наступну.

```

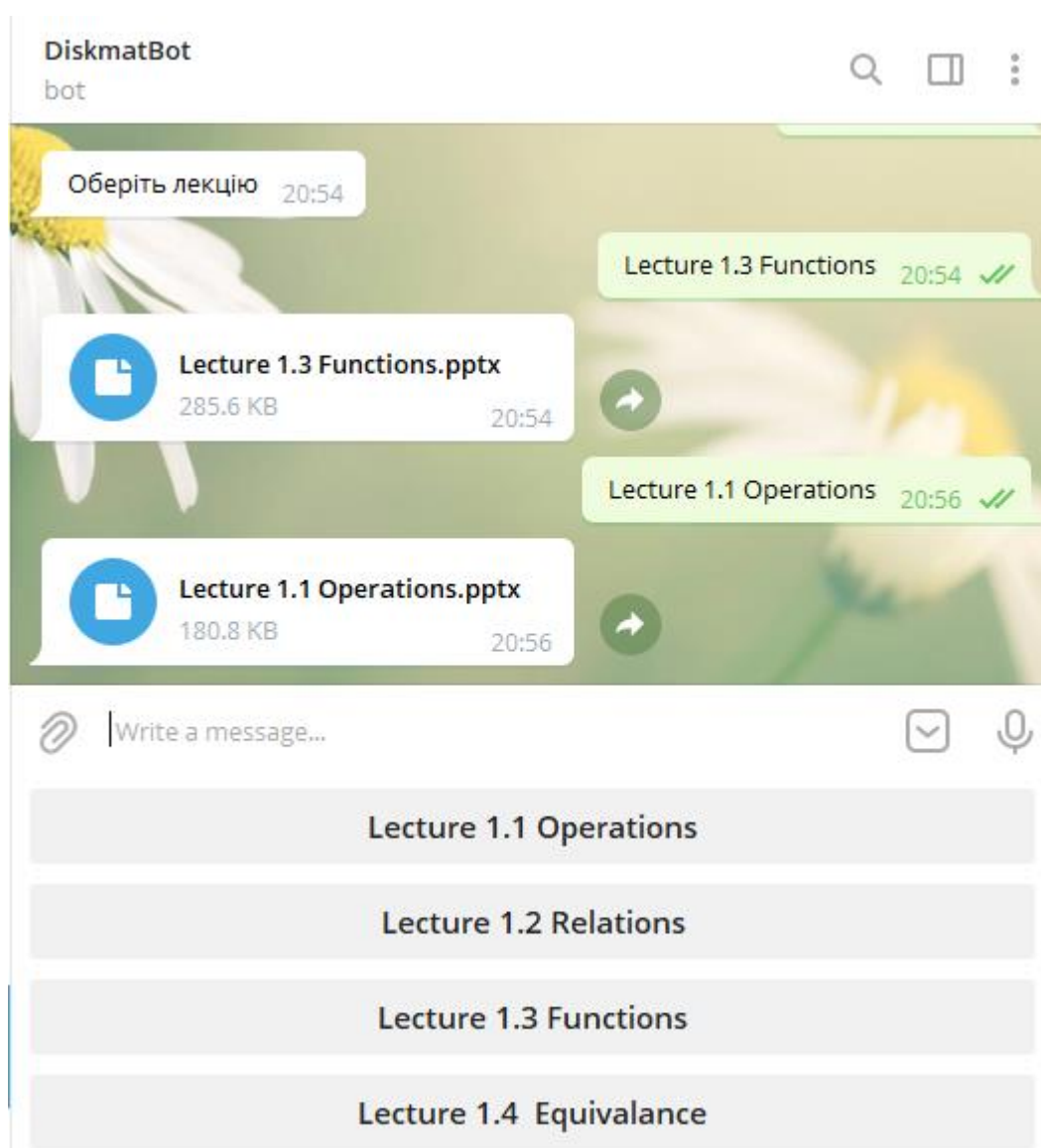
*/
protected function longpoll() {
    $params = array(
        'limit' => $this->updatesLimit,
        'timeout' => $this->updatesTimeout,
    );
    if ($this->updatesOffset) {
        $params['offset'] = $this->updatesOffset;
    }
    $options = array(
        'timeout' => $this->netConnectTimeout + $this->updatesTimeout + 2,
    );
    $response = $this->request('getUpdates', $params, $options);
    if ($response['ok']) {
        $updates = $response['result'];
        if (is_array($updates)) {
            foreach ($updates as $update) {
                $this->updatesOffset = $update['update_id'] + 1;
                $this->onUpdateReceived($update);
            }
        }
    }
    $this->longpoll();
}

abstract public function onUpdateReceived($update);
}

```

Отже, створено Телеграм бот DiskmatBot. Основними функціями бота є надсилання студентам навчальних матеріалів, отримання зворотного зв'язку від студентів. Користувачі можуть з допомогою меню, що реалізоване у вигляді кнопок, переходити до тих чи інших частин бота. Можна переглядати і завантажувати презентації, завдання для практичних робіт, отримувати завдання для контрольних робіт, ознайомлюватись з журналом відвідування та успішності. Користувацький інтерфейс представлено на рис. 4.1.

Рисунок 4.1 – Інтерфейс користувача в Телеграм-боті DiskmatBot



ВИСНОВКИ

Рівень популярності соціальних мереж та месенджерів постійно зростає. Це пов'язано з розвитком технологій та зручністю онлайн-спілкування. Також за останній рік велика кількість людей перейшла на дистанційну роботу. У час дистанційної освіти месенджери дозволяють покращити комунікацію між студентами та викладачами. Студентам зручніше користуватися системою Телеграм, ніж, наприклад, електронною поштою.

У ході роботи було досліджено інформаційну систему Телеграм, принцип роботи і можливості Телеграм ботів. Досліджено існуючі боти, їх функціонал. На основі дослідженні і потреб навчального закладу поставлено завдання створити Телеграм-бот для дистанційного курсу «Дискретна математика» на факультеті Комп'ютерних наук та кібернетики Київського національного університету імені Тараса Шевченка. Визначено функціональні та нефункціональні вимоги створюваного продукту.

В рамках випускної кваліфікаційної роботи на здобуття ступеня бакалавра було досліджено різні платформи для створення Телеграм-бота та обрано мову програмування PHP, cURL, json, long polling. Створено бота і наповнено його лекційними та практичними матеріалами з дискретної математики. Запущено бот на сервері Факультету комп'ютерних наук та кібернетики. Таким чином, студенти можуть у зручній для них формі отримувати доступ до навчальних матеріалів.

Дана робота може бути використана під час навчального процесу для поліпшення зручності отримання студентами факультету навчальних матеріалів. Студенти активно користуються месенджером Телеграм для обміну повідомленнями. Старости груп надсилають файли з навчальною інформацією, розкладом, лабораторними роботами та інші файли саме в групові чати Телеграм. Такі повідомлення швидко губляться через велику кількість учасників і повідомлень у чаті.

Варто продовжувати дослідження і роботу за цією тематику. Оскільки об'єми інформації швидко зростають, з'являється потреба у їх сортуванні та зручному представленні. Для цього використовують різні методи та засоби, в тому числі чат-боти. Планується збільшення функціоналу в розробленому Телеграм-боті. Це такі можливості, як проведення тестових контрольних робіт, проведення опитувань, вікторин.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Bots: An introduction for developers [Електронний ресурс]. Режим доступу до ресурсу: <https://core.telegram.org/bots>.
2. MDN Web Docs - JavaScript [Електронний ресурс]. Режим доступу до ресурсу:
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction#what_is_javascript.
3. telegraf.js - v4.3.0 [Електронний ресурс]. Режим доступу до ресурсу: <https://telegraf.js.org/>.
4. telegraf v4.3.0 [Електронний ресурс]. Режим доступу до ресурсу: <https://telegraf.js.org/>.
5. Що таке MongoDB? [Електронний ресурс]. Режим доступу до ресурсу: <https://www.mongodb.com/what-is-mongodb>.
6. PHP.NET [Електронний ресурс]. Режим доступу до ресурсу: <https://www.php.net/docs.php>.
7. Примеры использования JSON-формата на PHP и JavaScript [Електронний ресурс]. Режим доступу до ресурсу: <https://codernotes.ru/articles/php/json-format-na-primere-php-javascript.html>.
8. Бардзелл Джеффри Macromedia Dreamweaver MX 2004 с ASP, ColdFusion и PHP. Из первых рук (+ CD-ROM); Эком - М., 2016. - 560 с.
9. Зандстра Мэтт PHP. Объекты, шаблоны и методики программирования; Вильямс - М., 2016. - 560 с.
10. telegram-poll-bot [Електронний ресурс]. Режим доступу до ресурсу: <https://github.com/kolar/telegram-poll-bot/blob/master/TelegramBot.php>.