МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ВИПУСКНА РОБОТА

на тему:

«Інформаційне та програмне забезпечення телеграм бота надання першої медичної допомоги»

Завідувач випускаючої кафедри — **Довбиш А.С.**

Керівник роботи — **Шаповалов С.П.**

Студентки групи ІН – 61 — **Іванової А.О.**

СУМИ 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютерних наук

Затверджую_____

Зав. кафедрою Довбиш А.С.

"_____" _____2020 р.

ЗАВДАННЯ
до випускної роботи

Студентки четвертого курсу, групи ІН-61 спеціальності "Інформатика" денної форми навчання Іванової Анни Олегівни.

Тема: " Інформаційне та програмне забезпечення телеграм бота надання першої медичної допомоги"

Затверджена наказом по СумДУ

№_____ от _____2020 р.

Зміст пояснювальної записки: 1) аналіз проблеми та постановка задачі; 2) вибір метода розв'язання задачі; 3) розробка інформаційного і програмного забезпечення системи

Дата видачі завдання "_____"_____2020 р.

Керівник випускної роботи _____ Шаповалов С.П.

Завдання прийняла до виконання _____Іванова А.О.

РЕФЕРАТ

Записка: 60 стор., 53 рис., 2 табл., 1 додаток, 7 джерел.

Об’єкт дослідження — чат бот для месенджера Telegram**.**

Мета роботи — інформаційне та програмне забезпечення чат боту надання першої невідкладної допомоги для месенджера Telegram.

Методи дослідження — технології створення чат ботів.

Результати — розроблено інформаційну систему з надання першої невідкладної допомоги у вигляді чат боту для соціальної мережі Telegram. Створений бот зручний у користуванні, має декілька меню для пришвидшення отримання доступу до інформації. Розробка проводилась на базі мови програмування C#. У ході тестування проблем не виявлено.

TELEGRAM, BOT, C#, MEDICINE, HELP, MENU, DANGER, HURT

ЗМІСТ

ВСТУП 5

1 АНАЛІЗ ВІДОМИХ РІШЕНЬ 7

1.1 Огляд схожих інформаційних систем 7

1.2 Постановка задачі 15

2 ВИБІР ОСНОВНИХ КОМПОНЕНТІВ БОТУ ТА ІХ РЕАЛІЗАЦІЯ 16

2.1 Створення MindMap для проектування боту 16

2.2 Реєстрація бота в системі 19

3 КОМП'ЮТЕРНА РЕАЛІЗАЦІЯ ПРОЕКТУ ТА ТЕСТУВАННЯ 25

3.1 Вибір середовища розробки 25

3.2 Програмна реалізація основного функціоналу 26

3.4 Розміщення Telegram бота на сервері 29

3.4 Тестування Telegram боту 32

ВИСНОВКИ 40

СПИСОК ЛІТЕРАТУРИ 41

ДОДАТОК 42

ВСТУП

В сучасному світі, де все постійно модернізується людина постійно опановує нові гаджети. Сьогодення важко представити без новітніх технологій, не існує жодної сфери роботи де би не використовувались інформаційні технології : в школах використовують електронні щоденники та книжки, а під час карантину й сама школа перетворилася в онлайн навчання, навіть така людська професія як водій скоро може зникнути завдяки створенню безпілотних автомобілей, які вже почали використовувати.

Завдяки розвитку інтернет технологій кожного дня з'являються все нові можливості для медицини. Однією із таких можливостей наприклад є фітнес браслети завдяки яким можна контролювали серцебиття та показники артеріального тиску собі під час занять спортом або іншій людині навіть із віддаленої точки світу через мобільний додаток. Дуже часто такі додатки містять в собі спеціальні програми – боти, які імітують живе спілкування. Таке спілкування істотно економить час, що є величезним плюсом для людей, які стараються правильно розподіляти свій час.

Багато додатків розташовано в просторах інтернету і якщо скачувати їх із неперевірених сайтів можна занести до телефону злочинні віруси. Повільна реєстрація із постійною рекламою та додатки з мінімально корисним функціоналом також не є фаворитами користувачів. Тому є потреба у створенні корисних додатків із багатим функціоналом та розташуванням їх на вже роками перевірених платформах.

Саме тому мною для простору роботи мого бота було вибрано саме месенджер Telegram. На перший погляд це стандартний додаток для обміну повідомленнями, проте він не містить нав'язливої реклами, має великий ступінь захисту та розповсюджений в нашій країні. Також суттєвим є досить гарна та зрозуміла документація та наявність готових бібліотек, що значно пришвидшує розробку. Моєю ціллю є створення такого боту, користуючись яким людина за мінімальний проміжок часу отримає інформацію як надати

першу медичну допомогу(далі ПМД). При цьому інформація має містити всі необхідні рекомендації та поради, щоб навіть людина яка робить це вперше справилась без проблем.

1 АНАЛІЗ ВІДОМИХ РІШЕНЬ

Для створення ефективної інформаційної системи по наданню ПМД важливо оглянути вже відомі рішення щоб запозичити корисні і відмовитись від неефективних сторін цих систем. Також такий аналіз допоможе скласти оптимальний план ефективної роботи.

1.1 Огляд схожих інформаційних систем

1.1.1 Мобільний додаток для навчання і надання першої медичної допомоги

Перше, що кидається у очі у данному додатку - це розділ підготовки. У ньому перелічені екстремальні ситуації (далі ЕС), такі як: землетрус, пожежа, пандемія, ураган. Це допомагає людині у вільний час навчитися поводитися в тій чи іншій ЕС (рисунки 1.1-1.2).

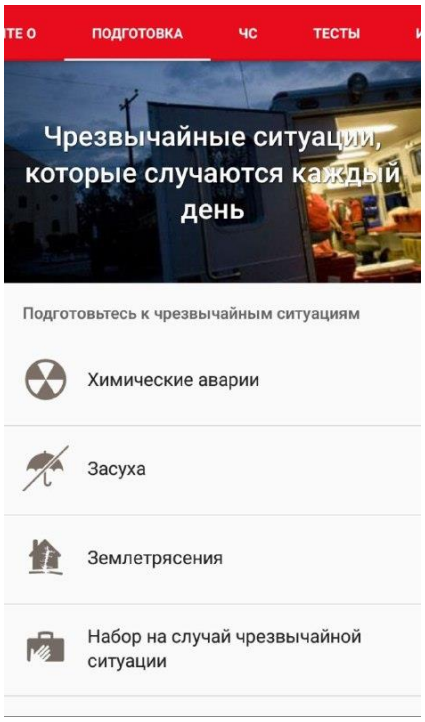

Рисунок 1.1 - Розділ підготовки до ЕС

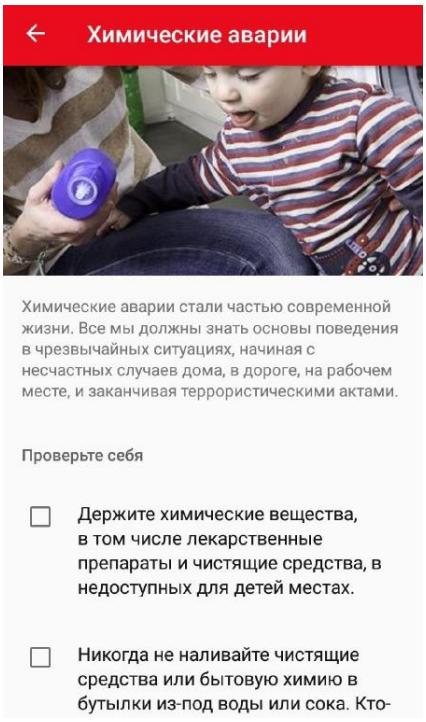

Рисунок 1.2 - Вкладка теми Хімічних аварій

Не менш цікавим є розділ тест, де кожен користувач може перевірити набуті ним знання (рисунок 1.3).

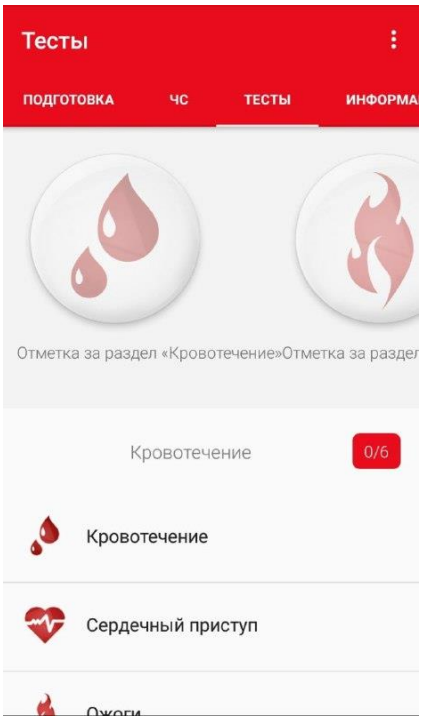

Рисунок 1.3 - Розділ «Тест» для перевірки знань

Якщо ж у вас не було можливості підготуватися завчасно та закріпити знання теоретично в нагоді стане розділ ЕС.

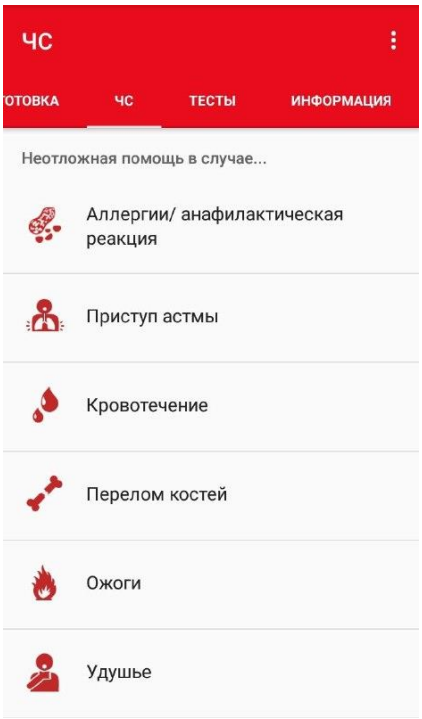

Рисунок 1.4 - Розділ ЕС

Він надасть усю необхідну інформацію для швидкої невідкладної допомоги в різних ситуаціях. Наприклад, при відкритті теми алергії ви побачите інформаційну сторінку як зображено на рисунку 1.5.

Рисунок 1.5 - Вкладка теми алергії

Можна побачити, що інформація гарно структурована. Спочатку розміщена весь основний текст, нижче, якщо щось незрозуміло, створений розділ запитань із відповідями з цієї теми, що дозволяє ознайомитись тільки з необхідною вам інформацією (рисунок 1.6).

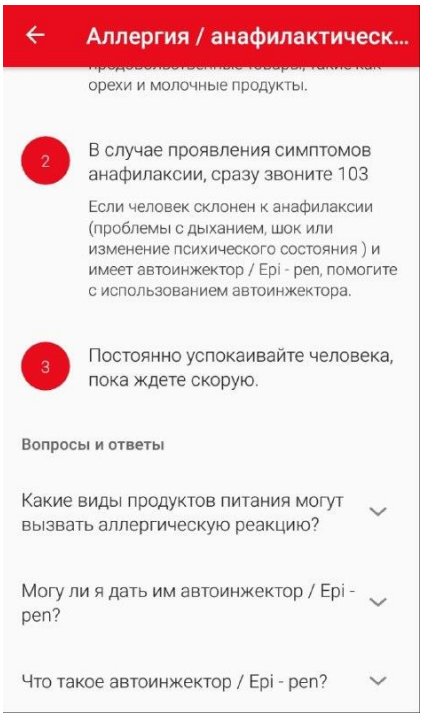

Рисунок 1.6 - Вкладка теми алергії, запитання та відповіді

Додаток інтуїтивно зрозумілий та дозволяє швидко включитись в роботу. На головній сторінці продубльовано теми з розділу ЕС в розширеному варіанті.

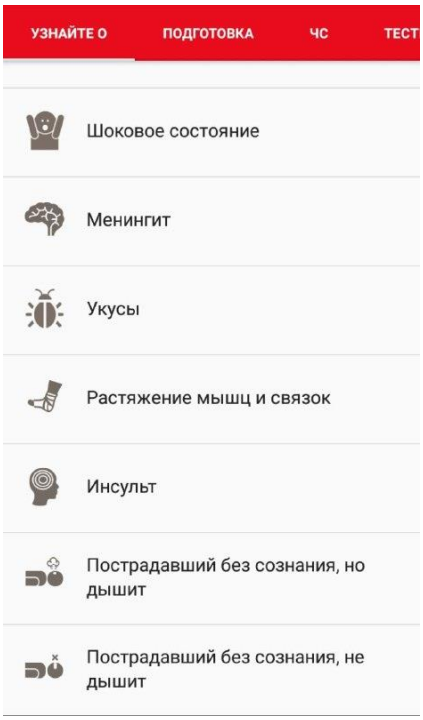

Рисунок 1.7 - Головна сторінка додатку

Однією із головних переваг цього додатку є офлайновість, тобто він працює навіть за відсутності з'єднання з мережею інтернет. Негативною стороною є використання пам'яті телефону, що є суттєвим для людей з старими моделями телефонів та людей які звикли використовувати кожен Мб пам'яті пристрою. Ще однією з негативних сторін додатку є час затрачений на його встановлення, що може коштувати комусь життя[1].

1.1.2 Месенджер-бот для першої медичної допомоги при передозуванні психоактивними речовинами

При роботі з данним ботом перевага офлайновості, яка була доступна в розглянутому вище мобільному додатку зникає, адже бот не може функціонувати без мережі інтернет. Проте користувач має змогу переглядати старі повідомлення в чаті при відсутності підключення до інтернету.

Початок роботи будь-якого бота зазвичай починається з команди /start або з кнопки почати (рисунок 1.8). Telegram створив для своїх користувачів захист від нав'язливої реклами – бот ніколи не може написати першим, він тільки відповідає на запит, який був до нього направлений. Це працює навіть у групових чатах, якщо не звертатися напряму до бота через його ім'я @bot, він не буде зчитувати повідомлення.

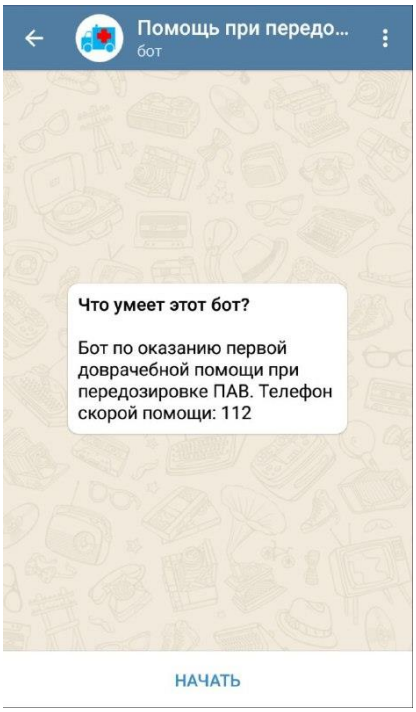

Рисунок 1.8 - Початок роботи з ботом

Бот має зрозуміле та зручне у користуванні меню що дозволяє економити час.

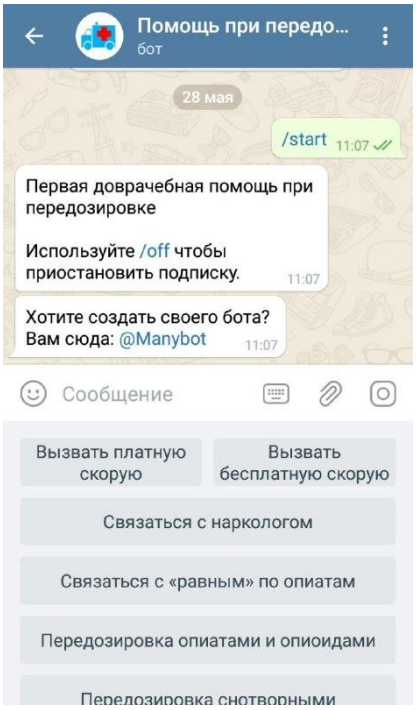

Рисунок 1.9 - Меню бота

При виборі відповідного пункту меню користувачеві відсилається вся необхідна інформація.

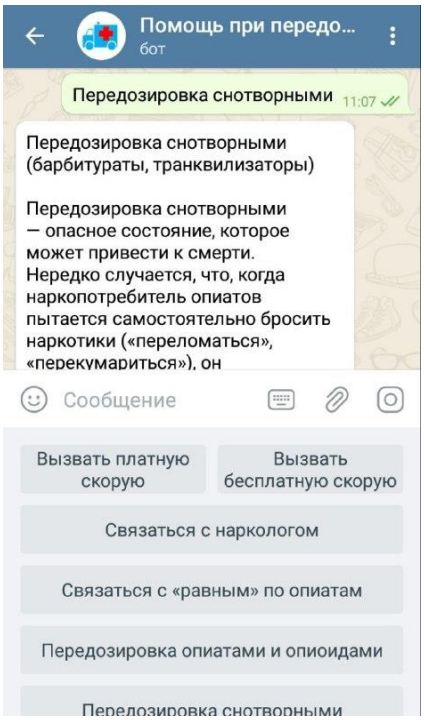

Рисунок 1.10 - Інформаційне повідомлення після вибору пункту передозування

Також до відповідного повідомлення, у разі потреби, прикріплена додаткова інформація, якої може потребувати користувач.

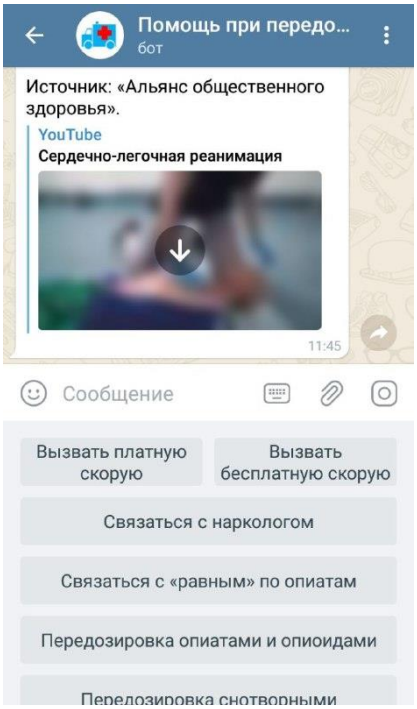

Рисунок 1.11 - Інформаційне повідомлення із відео контентом

Контент – інформаційний зміст матеріалів, сайтів, книжок тощо.

Інформаційний розділ «Про нас» надає всю актуальну інформацію про розробників проекту та сам проект[2].

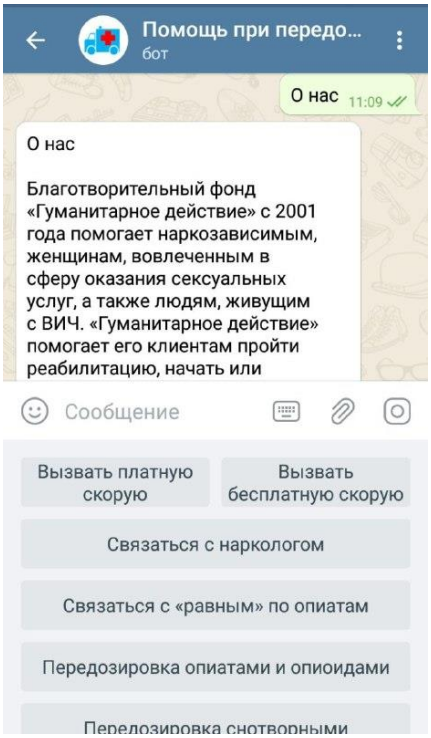

Рисунок 1.12 - Інформаційний блок "Про нас"

У результаті аналізу двох інформаційних систем зі схожою тематикою, було складено порівняльну таблицю параметрів, що допоможе під час проектування проекту.

Таблиця 1.1 Порівняльна характеристика параметрів інформаційних систем

Параметри	Додаток	Бот
Підготовка до роботи	Потрібно завантажити та встановити.	Знайти в пошуковому вікні месенджера.
Підготовка до ситуації	Присутня можливість завчасно дізнатись інформацію по тій чи іншій надзвичайній ситуації та закріпити знання тестуванням.	Окремого розділу нема, проте можливо завчасно ознайомитись з інформацією.
Зручність подання інформації	Інформація поділена на блоки. При натисканні на один із блоків,	Інформація поділена на блоки та представлена у вигляді меню, після

	доступ до інших зникає.	натискання відповідної кнопки доступ до інших залишається.
Доступ до інформації офлайн	Додаток коректно працює без доступу до мережі інтернет.	Не працює без інтернету, доступ до попередніх повідомлень(інформації) залишається.
Можливість зв’язатись з розробником	Відсутня.	Окремий пункт меню.

1.2 Постановка задачі

У результаті проведеного аналізу було виділено мету цієї роботи - проектування, розробка та програмна реалізація телеграм боту зі зручним інтерфейсом користувача та швидким доступом до всієї інформації. Для здобуття цієї мети необхідно завершити наступні завдання:

1) Вивчення теоретичних аспектів програмування чат боту на базі Telegram;
2) Проектування зв’язків;
3) Реєстрація бота в системі;
4) Розробка наповнення боту;
5) Завантаження боту на сервер
6) Провести тестування сторонніми користувачами;

2 ВИБІР ОСНОВНИХ КОМПОНЕНТІВ БОТУ ТА ІХ РЕАЛІЗАЦІЯ

2.1 Створення MindMap для проектування боту

MindMap це така діаграма зв’язків, або її ще називають асоціативною картою яка допомагає структурувати ідеї за допомогою графічного запису у формі діаграми. Вона представляє собою деревовидну структуру та відображає зв’язки між фрагментами інформації.

Створюючи MindMap я хочу спроектувати логіку бота для простоти програмування.

За основу своєї карти я взяла команди на які буде відповідати бот (рисунок 2.1).

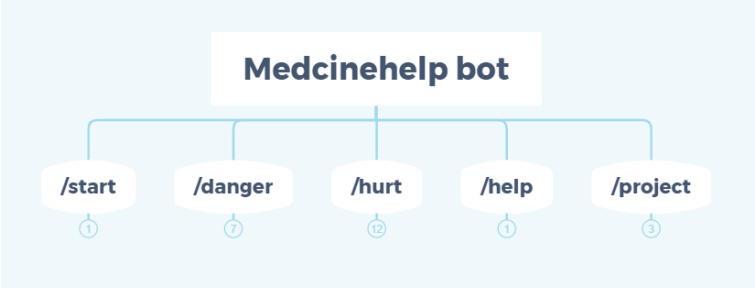

Рисунок 2.1 - Основа діаграми зв’язків

Після початку роботи із ботом користувач отримає вітальне повідомлення, яке розповість про бот та команди для роботи з ним.

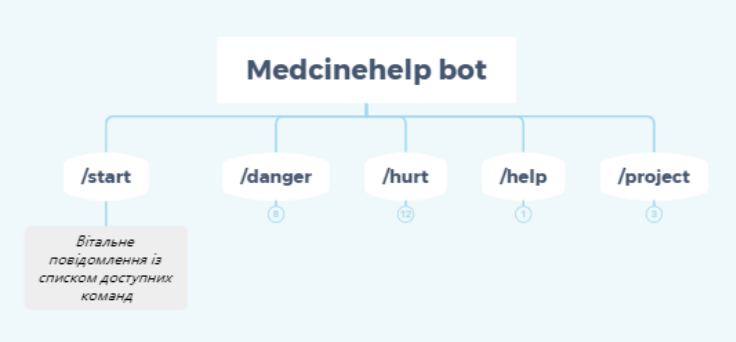

Рисунок 2.2 – Команда /start

При натисканні команди /danger користувач зможе побачити меню загроз, загрози в якому були розподілені на 6 частин (рисунок 2.3):

- Можливий обвал бурульок, цеглин тощо.
- Можливий наїзд автомобілем
- Можливий удар електричним струмом
- Можливий напад собак
- Можливий напад на вас людиною, якій збираєтесь допомогти
- Загроз не виявлено

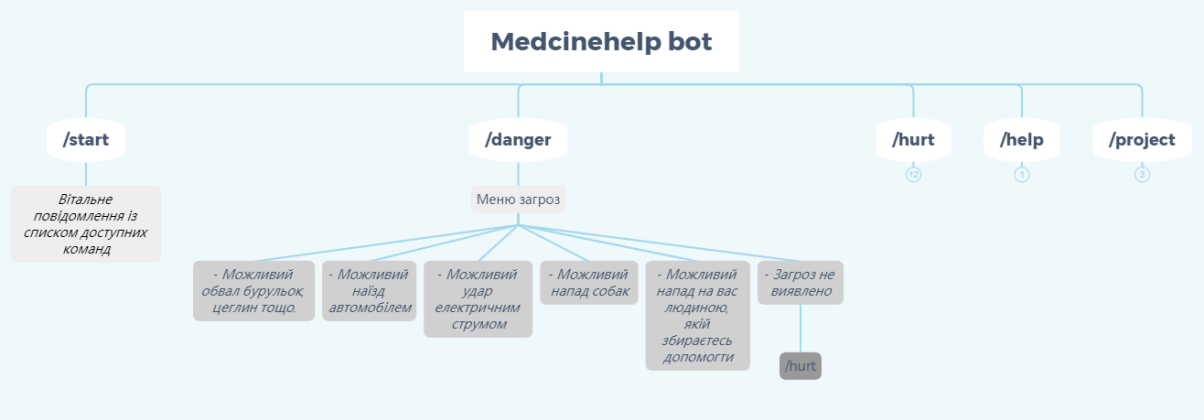

Рисунок 2.3 – Меню загроз

Для кожної кнопки я розробила текст як уникнути небезпеки в цій чи іншій ситуації. Наприклад, якщо було натиснуто на кнопку «Можливий удар електричним струмом» бот відправить текстове повідомлення до якого прикріплено відео як проводити серцево-легеневу реанімацію:

«Якщо поряд із потерпілим розлита вода, чи інша рідина яка проводить струм, не підходьте.

Якщо місце поряд із людиною сухе ви можете підійти і звільнити людину від дії струму. Існує 2 варіанти вирішення проблеми:

-Взяти суху палицю, дощечку або інший предмет що не проводить струм і звільнити постраждалого від струмовивідних частин/загроз;

-Можна відтягнути постраждалого за його одяг якщо він сухий та відстає від тіла;

Відтягніть людину від небезпечної частини щонайменше на 8-10 метрів

В разі необхідності зробіть штучне дихання і напоїть людину водою та чекайте на приїзд швидкої допомоги.

https://youtu.be/CHp8MTpkocg ».

В свою чергу після натиску кнопки «Загроз не виявлено» вам прийде повідомлення із командою, яка надасть доступ до списку можливих ушкоджень (рисунок 2.4).

«Тепер, коли загроза життю не актуальна можна допомогти людині. Виберіть вид ушкодження із меню ушкоджень /hurt».

Для команди /hurt було розроблено відповідне меню ушкоджень яке стане доступне разом із супроводжуючим повідомленням («Виберіть тип ушкодження»).

Після натискання команди ви побачите меню(рисунок 2.4) із такими пунктами[3]:

- Поранення
- Кровотеча
- Перелом
- Перелом голови
- Сонячний урар
- Обмороження
- Утоплення
- Укуси:
    - Укуси отруйних змій
    - Укуси тварин
    - Укуси комах

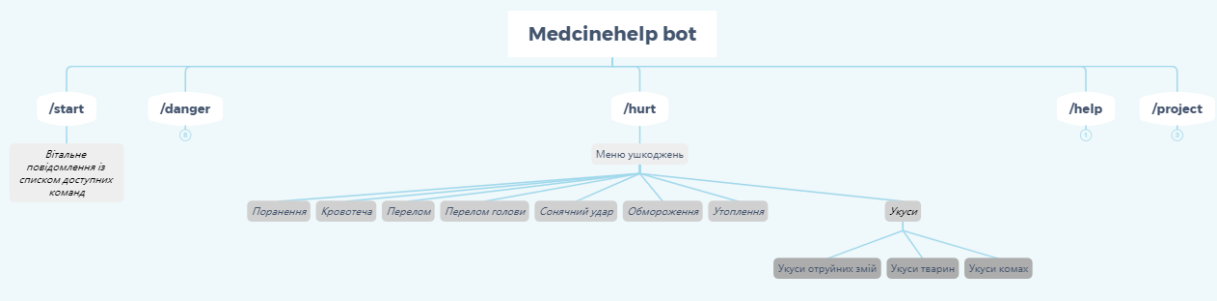

Рисунок 2.4 – Меню ушкоджень

Після вибору вам буде надано актуальну інформацію щодо дій.

Таким чином я проаналізувала і систематизувала всю інформацію яку необхідно знати людині що надає ПМД. Окрім інформаційний меню я створила ще меню проекту(рисунок 2.5).

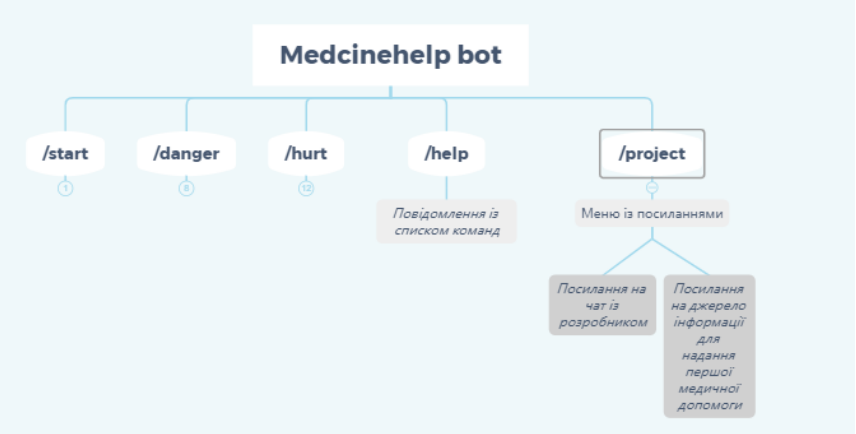

Рисунок 2.5 – Меню проекту

Команда ж /help відправляє користувачу список команд для маніпулювання ботом(рисунок 2.5)

2.2 Реєстрація бота в системі

Перш ніж приступати до роботи над програмною реалізацією бота в Telegram, є необхідним його зареєструвати. Що добре, сам цей процес дуже простий і повністю автоматизований.

В мережі Telegram для створення та реалізації ботів існує окремий аккаунт @BotFather, що допоможе вам при розробці власного боту. На рисунку 2.6 зображено початок роботи із цим ботом.

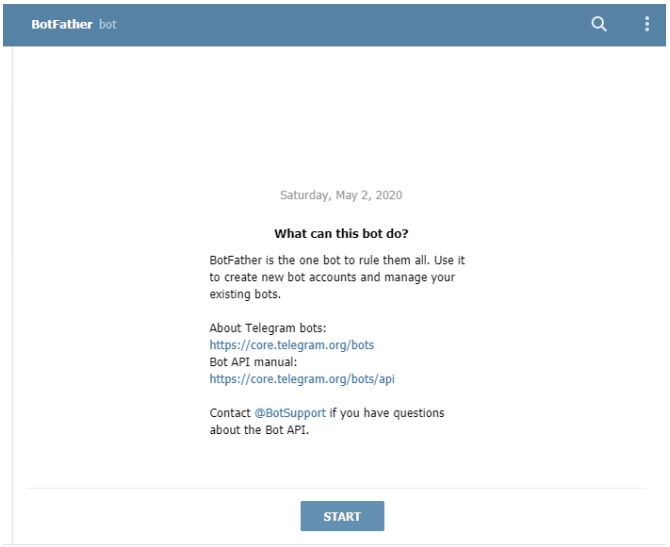

Рисунок 2.6 - Початок роботи з ботом-реєстратором

Після отримання списку базових команд для роботи із @BotFather натискаємо / newbot для створення нового боту (рисунок 2.7).

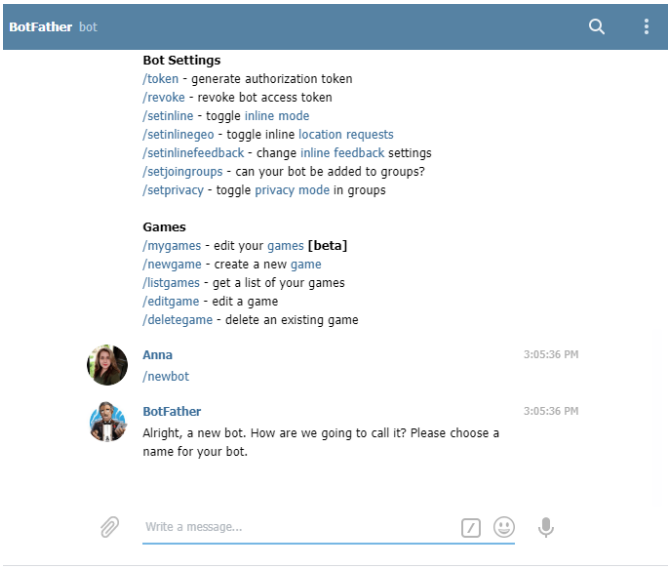

Рисунок 2.7 - Створення нового бота

Відправляєте бажане вами ім'я для нового бота і його юзернейм, який обов'язково повинен закінчуватися на "bot". Для свого проекту я вибрала назву Medcinehelp bot.

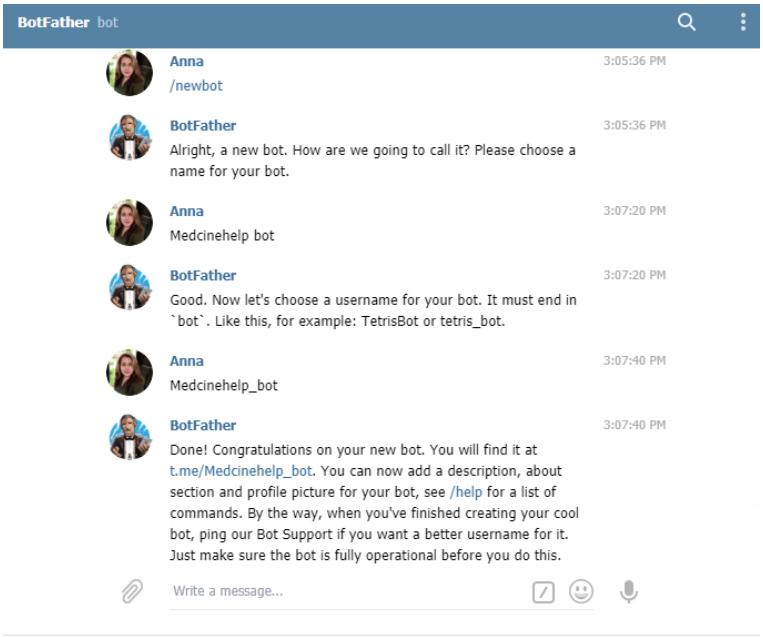

Рисунок 2.8 - Іменування бота

Отримуєте унікальний токен, що надасть вам доступ до HTTP API

Токен – це такий цифровий ключ, що допоможе вам пройти ідентифікацію в який-небудь важливий об’єкт або відкрити вашу банківську комірку. Його відмінність від звичайного пароля полягає в тому, що цифровий ключ весь час оновлюється(постійно генерує нові числа), він відомий лише вам, в момент ідентифікації.

HTTP – це протокол прикладного рівня моделі OSI, який використовується для передачі данних.

API – це набір готових інтерфейсів (класів, функцій, констант, процедур та структур), що надаються додатком (бібліотекою, сервісом) для використання в зовнішніх програмних продуктах.

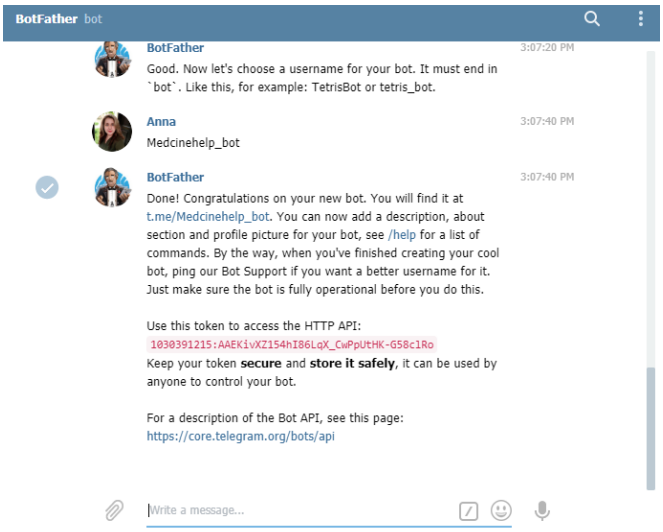

Рисунок 2.9 - Отримання токену для маніпулювання ботом

При бажанні, відразу можна ввести опис для бота, список його команд і відповідне зображення або відкласти ці дії на потім. На рисунку 2.10 зображено чат після вибору команди /setdescription , де я ввожу інформацію для опису боту «Цей телеграм бот швидко надасть вам всю необхідну інформацію для першої медичної допомоги»

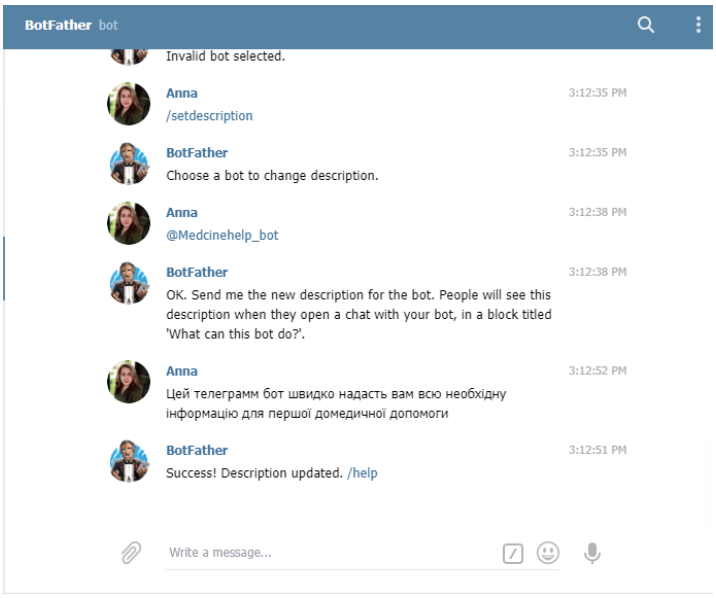

Рисунок 2.10 - Створення опису телеграм бота

Далі я ввожу /setcomands для створення списку команд (рисунок 2.11) та /setuserpic для завантаження фото профілю (рисунок 2.12).

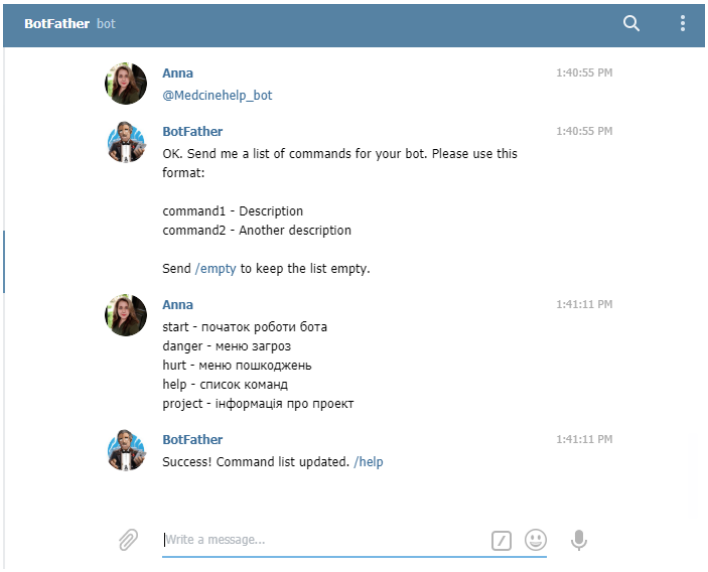

Рисунок 2.11 - Створення списку команд

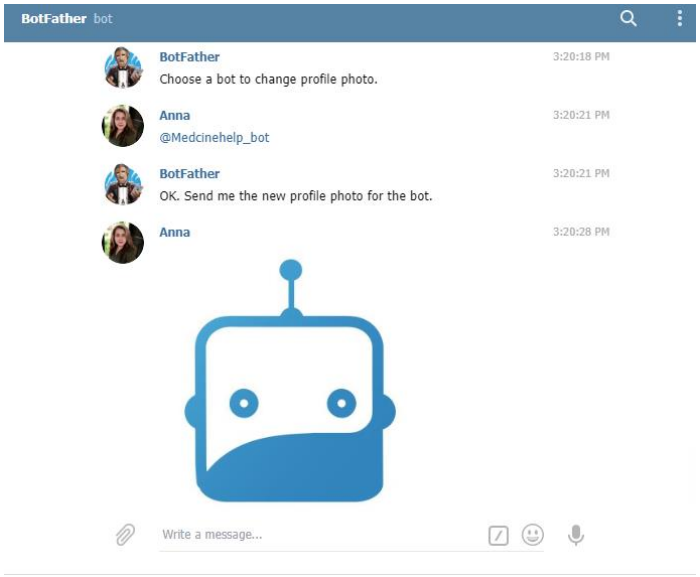

Рисунок 2.12 - Додавання фото до профілю бота

На рисунку 2.13 ми бачимо зображення готового профілю боту після вище зазначених дій.

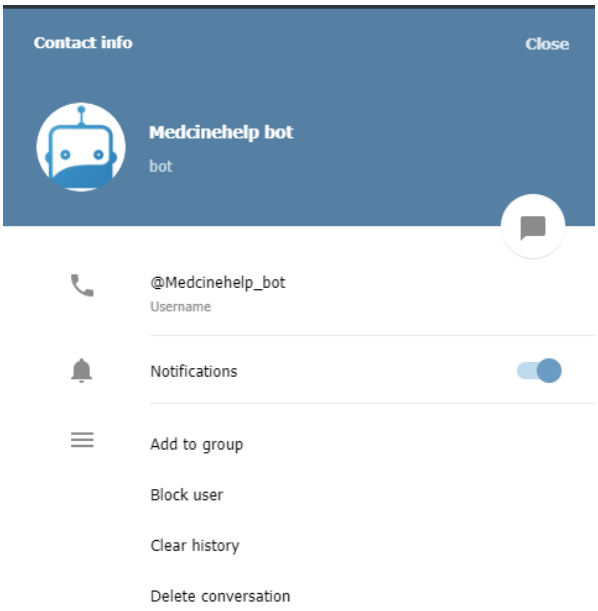

Рисунок 2.13 - Профіль боту

Таким чином ми отримали основу для боту із описом, фотографією та набором команд [4]:

- /start - початок роботи бота;
- /danger - меню загроз;
- /hurt - меню пошкоджень;
- /help - список команд;
- /project - інформація про проект;

3 КОМП'ЮТЕРНА РЕАЛІЗАЦІЯ ПРОЕКТУ ТА ТЕСТУВАННЯ

3.1 Вибір середовища розробки

Існує декілька шляхів втілення функціональності Telegram-бота. Кожен з них має свої недоліки та переваги, проте створення бота досить легко реалізовується так:

- Використовуючи Microsoft Visual Studio. При цьому мовою програмування доцільно вибрати C # або Node.js. Це дозволить пришвидшити розробку, адже писати багато коду буде не потрібно – простіше використовувати готовий набір шаблонів серед яких Bot Application, Bot Framework Emulator, Bot Dialog і Bot Controller. Уся необхідна інформація міститься в документації Microsoft Bot Framework. Програмісту потрібно тільки налаштувати шаблони під свої потреби та вимоги. На жаль, цим варіантом можна скористатися тільки користувачу операційної системи Windows.
- Іншим не менш корисним та зручним варіантом є мова програмування PHP. Він також дає багато можливостей для написання коду. Однією із головних переваг являється чимала популярність цієї мови – у разі непередбачуваної ситуації завжди знайдеться експерт, до якого можна звернутися. Наявність готових бібліотек, серед яких - Telegram Bot SDK, ще більше схиляє розробників на свою сторону, адже це дозволяє звести зусилля до мінімуму.
- Ну і звісно серед мов програмування, що дозволяють досить безпроблемно створити бота в телеграмі, Python є одним з найбільш лідируючих рішень. Перш за все це пов'язують з великими можливостями, які відкриваються при використанні стандартних бібліотек. Не менш важливим є можливість користуватися

спеціальними бібліотеками розрахованими на роботу з Telegram. Наприклад PyTelagramBotAPI яка бере на себе всі нюанси роботи, від відправки до отримання повідомлень.

Таблиця 3.1. Порівняльний аналіз мов програмування для створення ботів мережі Telegram

Параметри	C#	PHP	Python
Шаблони	+	-	+
Бібліотеки	+	+	+
Розповсюдженість мови	+	+	+
Читабельність	+	+	+

Звичайно, для того щоб забезпечити безперебійну роботу боту створення виявиться тільки половиною роботи. Іншим етапом стане підібрати та розмістити його на сервері. Існує чимало платних сервісів, що дозволить підібрати надійний за ціною та оптимальний сервер. Проте без проблем можна знайти і безкоштовні рішення такі як Azure або Heroku[5].

Я ж як постійний користувач Microsoft Visual Studio обираю саме C#, бо найбільш близьке та адаптоване середовище програмного забезпечення для вирішення поставлених задач. В якості хостингу на перший раз вирішила користуватись Deploy-F, так як він має гарну службу підтримки, що дозволить без проблем розмістити проект після завершення роботи.

3.2 Програмна реалізація основного функціоналу

Телеграм бот складається з файлу програмного Program.cs , в якому знаходяться методи для приймання, обробки та відправки повідомлень.

Окрім стандартних бібліотек в Program.cs також підключені бібліотеки Telegram Bot. Вони дозволяють організовувати зв’язок із користувачем та обробляти повідомлення.

```
using System;
using System.IO;
using System.Linq;
using System.Threading;
using System.Threading.Tasks;
using Microsoft.Extensions.Configuration;
using Telegram.Bot.Exceptions;
using Telegram.Bot.Extensions.Polling;
using Telegram.Bot.Types;
using Telegram.Bot.Types.Enums;
using Telegram.Bot.Types.InlineQueryResults;
using Telegram.Bot.Types.InputFiles;
using Telegram.Bot.Types.ReplyMarkups;
```

Рисунок 3.1 - Список підключених бібліотек до Program.cs

Для роботи з ботом необхідно створити об'єкт, що має тип TelegramBotClient.

```
private static TelegramBotClient Bot;
```

В головному методі присвоюємо цьому об’єкту унікальний ключ, який було отримано від @BotFather. Ключ записано у файл appsettings.json

```
public static async Task Main()
{

    IConfiguration configuration = new ConfigurationBuilder()
    .SetBasePath(Path.Combine(Directory.GetCurrentDirectory()))
    .AddJsonFile("appsettings.json", optional: true, reloadOnChange: true)
    .AddEnvironmentVariables()
    .Build();

    var botSettings = configuration.GetSection("BotSettings").Get<BotSettings>();

    var cts = new CancellationTokenSource();
```

Рисунок 3.2 – Частина методу Main

```
{
    "BotSettings": {
        "BotToken": "1030391215:AAEKivXZl54hI86LqX_CwPpUtHK-G58clRo"
    }
}
```

Рисунок 3.3 – Файл appsettings.json

Коли бот отримує повідомлення викликається функція Bot_OnMessageReceived() , в якій відбувається процеси обробки повідомлень, що написав користувач і повернення результату. Для обробки кнопок інлайн клавіатури використовуєть функція Bot_OnCallbackQueryReceived().

```
// Process Inline Keyboard callback data
private static async Task BotOnCallbackQueryReceived(CallbackQuery callbackQuery)
{

    var message = callbackQuery.Message;
    string buttonText = callbackQuery.Data;
    string name = $"{callbackQuery.From.FirstName} {callbackQuery.From.LastName}";
    Console.WriteLine($" {name} натиснув кнопку {buttonText}");
    if (callbackQuery.Data == "callback1")
    {
        await Bot.AnswerCallbackQueryAsync(callbackQuery.Id, "Відкриється чат із розробником");
    }
    else
        await Bot.AnswerCallbackQueryAsync(callbackQuery.Id, "Відкриється сслилка-джерело інформації по наданню домедичної допомоги");
}
```

Рисунок 3.4 - Метод Bot_OnCallbackQueryReceived

Для виводу назви боту використовується змінна me. Функція GetMeAsync() передає нам інформацію про наш бот, а саме назву боту (рисунок 3.5).

```
var me = await Bot.GetMeAsync();
Console.Title = me.Username;
Console.WriteLine("GetMe:" + me.Username);

Console.WriteLine($"Start listening for @{me.Username}");

await Bot.ReceiveAsync(new DefaultUpdateHandler(HandleUpdateAsync, HandleErrorAsync), cts.Token);
```

Рисунок 3.5 - Функція GetMeAsync()

В моєму чат боті я використовую 2 види клавіатур :

- ReplyKeyboard (рисунок 3.6)

  Кнопки цієї клавіатури обробляються в методі Bot_OnMessageReceived(). Саме вони використані для створення меню загроз, меню ушкоджень та меню укусів

```
var replyKeyboard = new ReplyKeyboardMarkup(new[]
{
    new[]
    {
        new KeyboardButton("Обвал бурульок, цеглин тощо.") ,
        new KeyboardButton("Наїзд автомобілем"),
    },
    new[]
    {
        new KeyboardButton("Удар елекричним струмом") ,
        new KeyboardButton("Напад собак"),
    },
    new[]
    {
        new KeyboardButton("Напад на вас людиною якій збираєтесь допомогти") ,
        new KeyboardButton("Загроз не виявлено"),
    },
```

Рисунок 3.6 - Створення ReplyKeyboard

- InlineKeyboard

  Кнопки цієї клавіатури обробляються в методі Bot_OnCallbackQueryReceived().

Їх я використовую для створення меню проекту.

```
var inlineKeyboard = new InlineKeyboardMarkup(new[]
{
    new []{
        InlineKeyboardButton.WithUrl("Зв'язатись з розробником","https://t.me/panda2804"),
        InlineKeyboardButton.WithUrl("Джерело медичної інформації","https://www.dzhmao.ru/info/inst
    },
    new []{
        InlineKeyboardButton.WithCallbackData("↑ info","callback1"),
        InlineKeyboardButton.WithCallbackData("↑ info","callback2"),
    }
```

Рисунок 3.7 - Створення InlineKeyboard

Після натискання кнопки із посиланням відкриється вікно із переходом на "Джерело медичної інформації".

```
private static async void Bot_OnCallbackQuery(object sd, CallbackQueryEventArgs ev)
{
    var message = ev.CallbackQuery.Message;
    string buttonText = ev.CallbackQuery.Data;
    string name = $"{ev.CallbackQuery.From.FirstName} {ev.CallbackQuery.From.LastName}";
    Console.WriteLine($" {name} натиснув кнопку {buttonText}");
    if (ev.CallbackQuery.Data == "callback1")
    {
        await Bot.AnswerCallbackQueryAsync(ev.CallbackQuery.Id, "Відкриється чат із розробником");
    }
    else
        await Bot.AnswerCallbackQueryAsync(ev.CallbackQuery.Id, "Відкриється сслилка-джерело інформації по наданню
}
```

Рисунок 3.8 - Обробка кнопок клавіатури InlineKeyboard

Після натискання інфо кнопки з’являється повідомлення у верхній частині чату.

3.4 Розміщення Telegram бота на сервері

Розміщення додатку на хостингу Deploy-F дуже просте та зрозуміле. Сервер не безкоштовний, проте для нових клієнтів надає 14 денний пробний період та невеличку початкову суму для платежу. Після швидкої реєстрації на сайті вам відкриється особистий кабінет в якому буде видно усі ваші проекти.

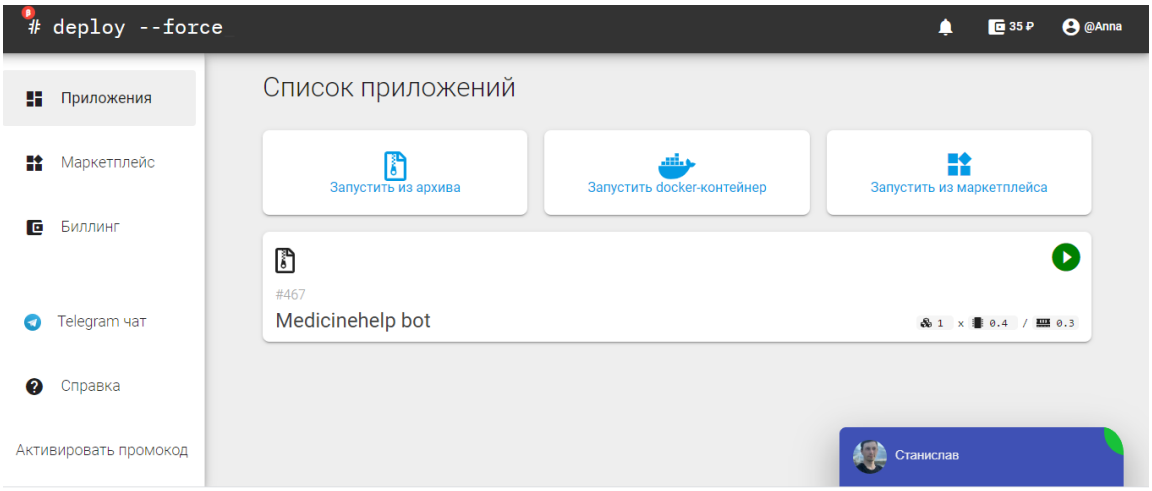

Рисунок 3.9 – Особистий кабінет користувача сайту Deploy-F

Як видно на рисунку 3.9 для завантаження доступні такі джерела як архів, докер контейнет та маркетплейс. Свого бота я завантажувала з архіву.

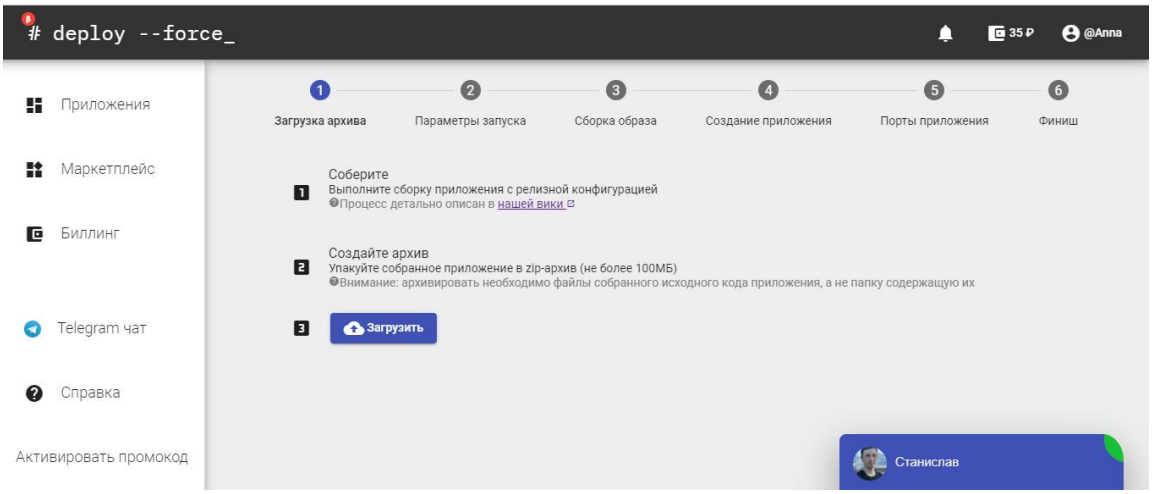

Рисунок 3.10 – Вкладка «Завантаження архіву»

Після завантаження архіву будуть доступні параметри запуску за замовчуванням(рисунок 3.11).

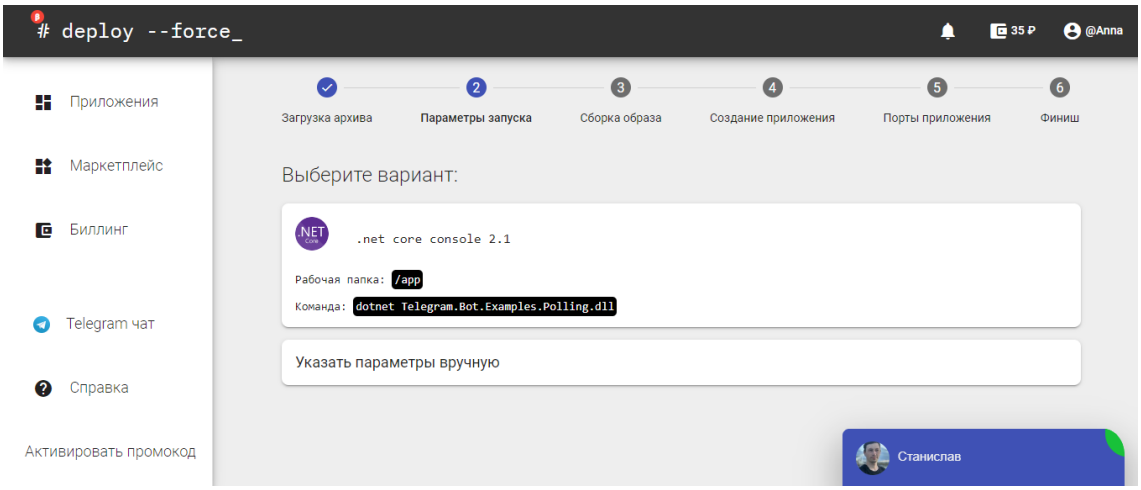

Рисунок 3.11 – Вкладка «Параметри запуску»

Після збірки образу ми створюємо віртуальний додаток, на цій же сторінці вписуємо його ім’я (рисунок 3.12).

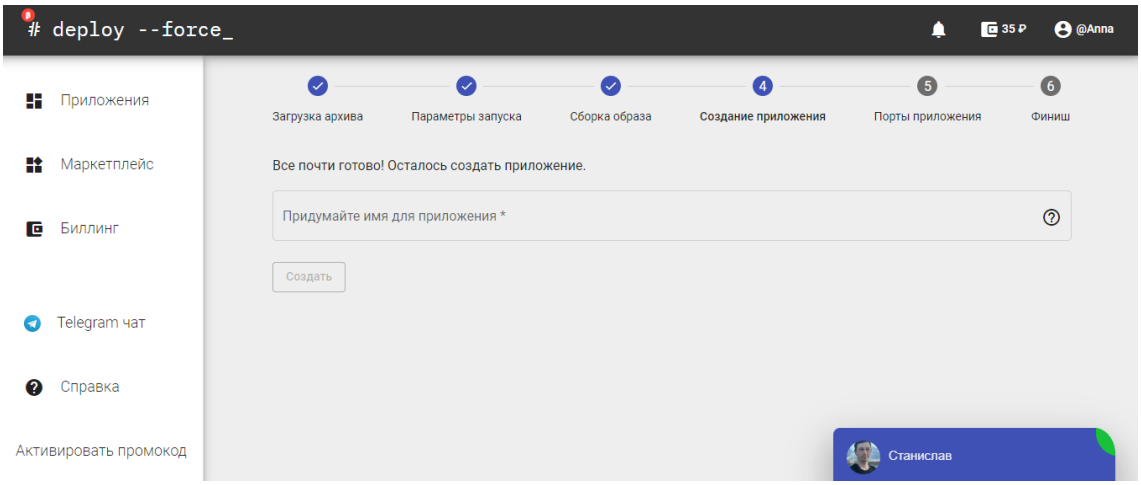

Рисунок 3.12 – Вкладка «Створення додатку»

На наступній сторінці «Порти додатку» ми можемо додади порти та проколи якщо іх використовує наш додаток, або просто натискаємо кнопку далі (рисунок 3.13).

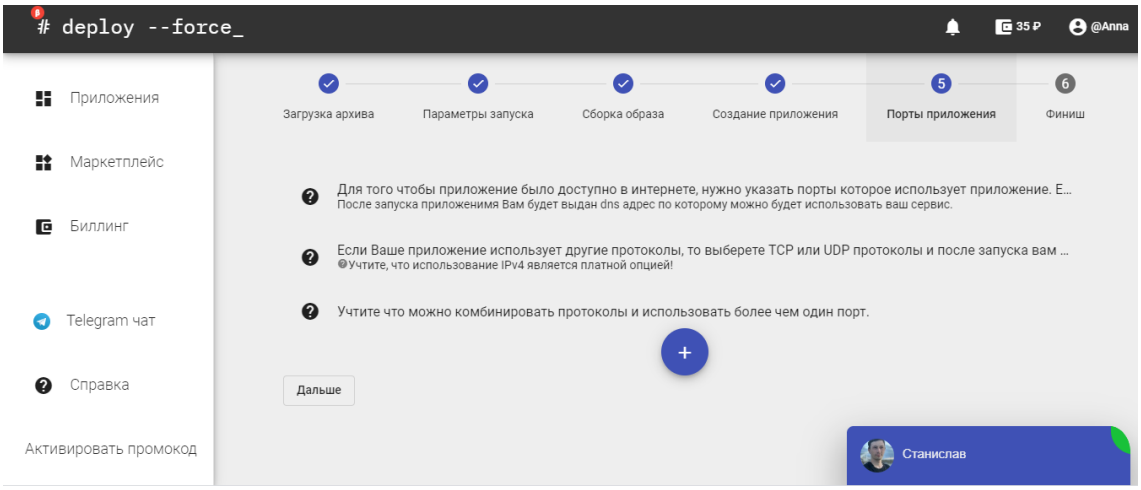

Рисунок 3.13 – Вкладка «Порти додатку»

Остання ж вкладка повідомляє нам про те, що наш віртуальний додаток створено (рисунок 3.14).

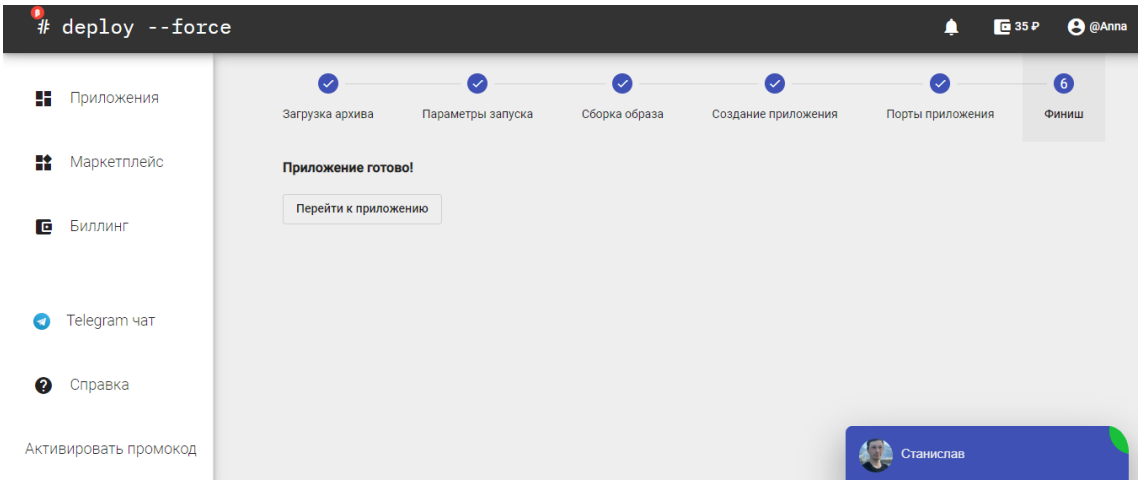

Рисунок 3.14 – Вкладка «Фініш»

3.4 Тестування Telegram боту

Після реалізації боту було проведення тестування різними сторонніми користувачами. В ході тестування проблем не виявлено.

На початку роботи бота користувач бачить опис того чим займається бот та кнопку «Начать» (рисунок 3.15), після натискання якої боту відправиться повідомлення із командою /start (рисунок 3.16).

Рисунок 3.15 – Початок роботи із ботом

На команду /start бот відповідає вітальним повідомленням із списком команд для керування ним (рисунок 3.16).

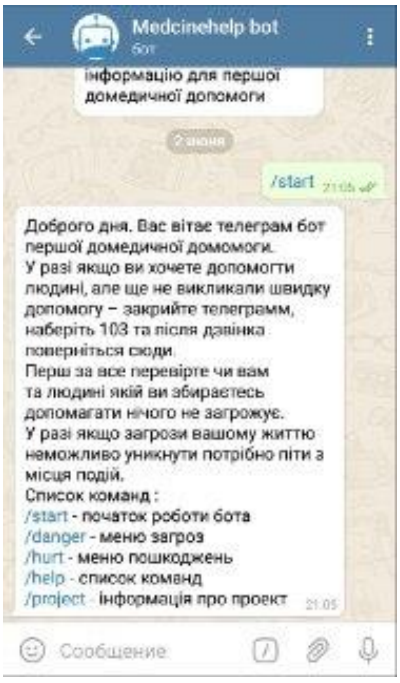

Рисунок 3.16 – Чат із ботом після команди /start

Якщо натиснути на слеш внизу екрану(рисунок 3.17) ми побачимо підказку із списком команд на які відповідає бот (рисунок 3.18)

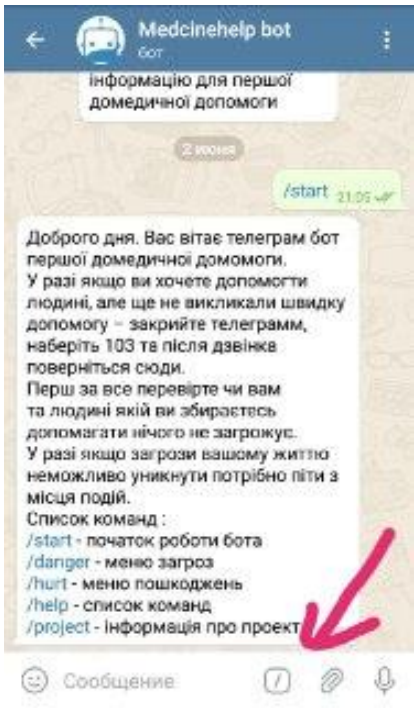

Рисунок 3.17 – Чат із ботом після команди /start перед натисканням «/»

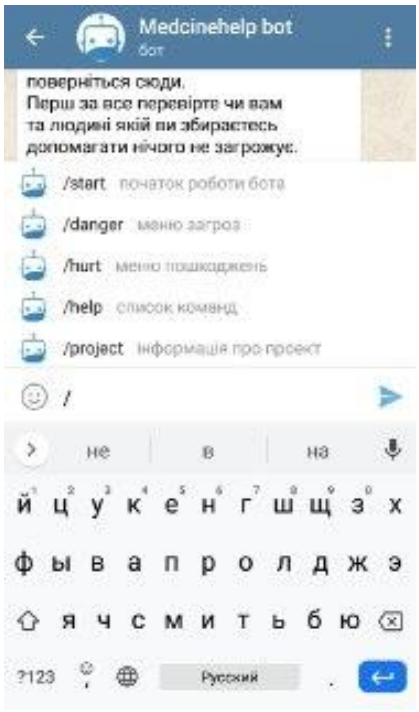

Рисунок 3.18 – Чат із ботом після команди /start та натискання «/»

Після натискання команди /danger можна побачити меню загроз. Якщо натиснути на одну із кнопок меню (рисунок 3.19), бот відправить повідомлення із інформацією як уникнути саме цієї загрози.

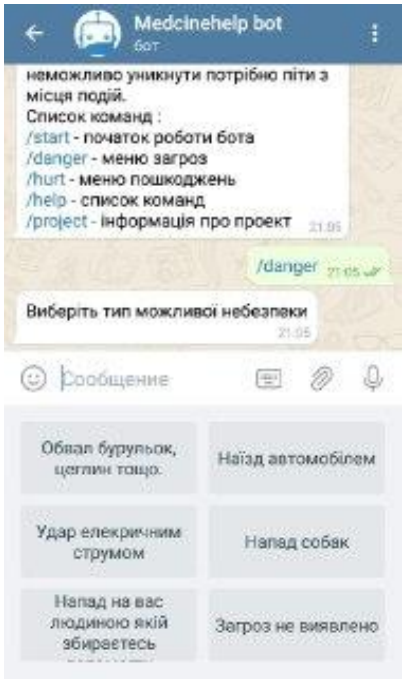

Рисунок 3.19 – Чат із ботом після команди /danger

Якщо у меню загроз натиснути кнопку «Напад собак» ми отримаємо таке повідомлення як на рисунку 3.20.

Рисунок 3.20 – Чат із ботом після натискання кнопки «Напад собак»

Після натискання команди /hurt ввідкриється відповідне меню ушкоджень із переліком можливих травм (рисунок 3.21).

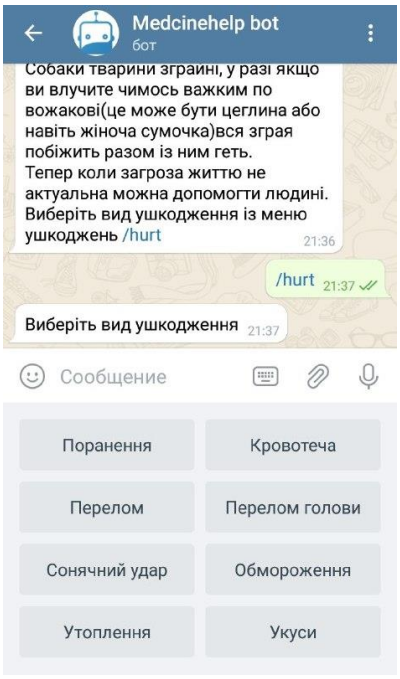

Рисунок 3.21 – Чат із ботом після команди /hurt

Після натискання будь-якого поранення ми побачимо інформацію по наданню допомоги при конкретному ушкодженні. Після натискання кнопки «Укуси» ми побачимо меню укусів, з якого можна вибрати саме той від якого постраждали (рисунок 3.22).

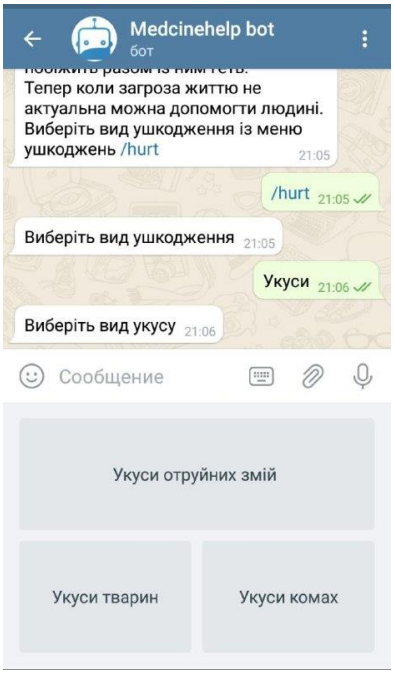

Рисунок 3.22 – Чат із ботом після команди /hurt

Після натиску кнопки «Укуси комах» ви отримаєте інформацію як надати допомогу в цьому випадку (рисунок 3.23). Також для більш повного ознайомлення прикріплено відео як робити серцево-легеневу реанімацію.

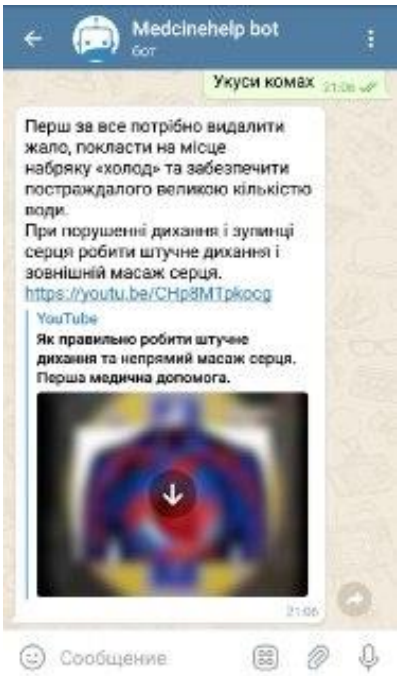

Рисунок 3.23 – Чат із ботом після натискання кнопки «Укуси комах»

Якщо ж натиснути команду /help, можна побачити список команд на які відповідає бот (рисунок 3.24).

Рисунок 3.24 – Чат із ботом після команди /help

Інформацію про проект можна дізнатись після натискання команди /project (рисунок 3.25).

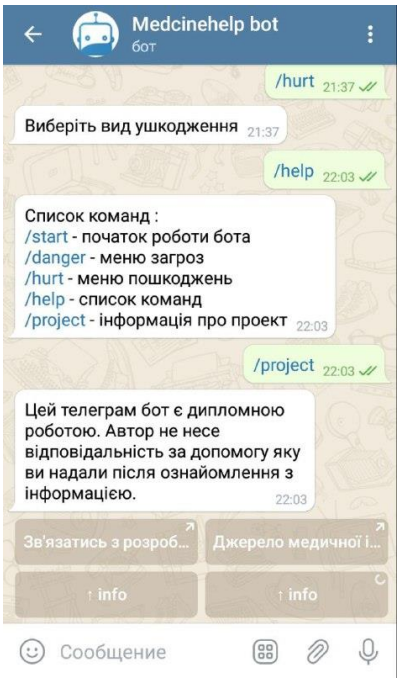

Рисунок 3.25 – Чат із ботом після команди /project

На рисунку 3.25 ми бачимо 4 кнопки 2 з яких із посиланнями, про що говорять маленькі стрілочки в правому верхньому кутку кнопок. Інші 2 кнопки це так звані інлайн кнопки після натискання яких повідомлення в чат не відправляються, а показуються в верхній частині екрану (рисунки 3.26-3.27).

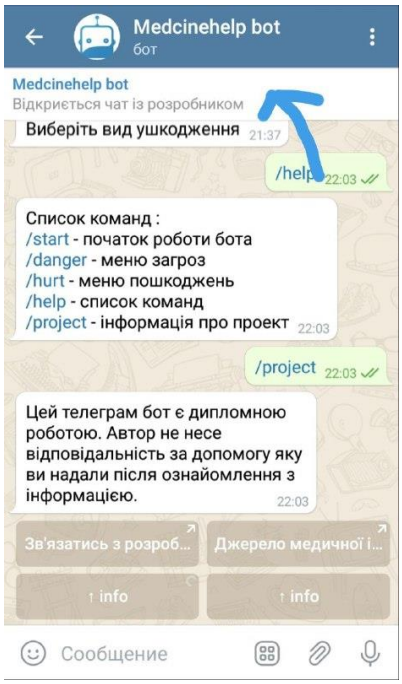

Рисунок 3.26 – Чат із ботом після натискання лівої інфо-кнопки

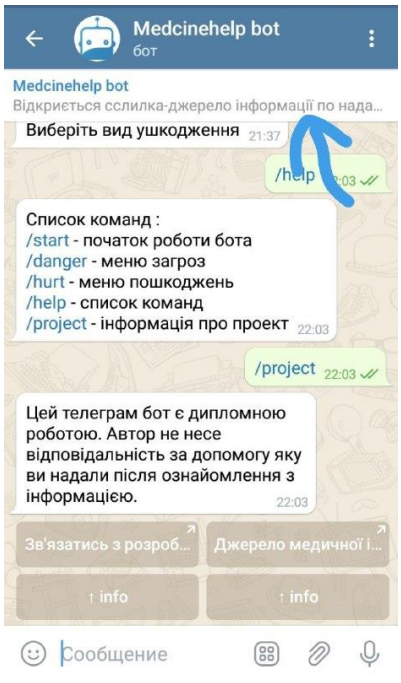

Рисунок 3.27 – Чат із ботом після натискання правої інфо-кнопки

Якщо натиснути на верхню кнопку-посилання побачимо вікно із запитом на відкриття чату із розробником або запитом на відкриття посилання-джерело медичної інформації (рисунок 3.28).

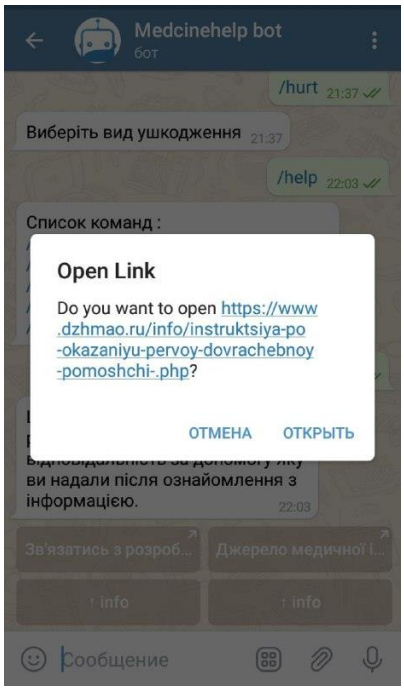

Рисунок 3.28 – Чат із ботом після кнопки із посиланням на джерело медичної допомоги

ВИСНОВКИ

В ході виконання роботи були вирішені наступні задачі:

- розглянуті та проаналізовані сучасні мови програмування для створення телеграм боту
- проведено аналіз тематичної літератури та огляд аналогічних ІС
- створено та налаштовано роботу боту
- проведено тестування працездатності боту

Результатом виконання роботи є створений функціональний телеграм бот, який допоможе вам надати ПДД.

Розроблений чат бот має ряд від’ємних рис:

- інтуїтивно зручний та зрозумілий інтерфейс
- команди доступу до різних меню
- зручні меню для швидкого ознайомлення з потрібною інформацією
- зв’язок із розробником в разі нагальної потреби

СПИСОК ЛІТЕРАТУРИ

1. Мобільний додаток для надання ПМД [Електронний ресурс] – Режим доступу: https://play.google.com/store/apps/details?id=com.cube.gdpc.fa&hl=ru – Дата доступу 1.04.2020
2. Telegram bot при передозировке психоактивными веществами [Електронний ресурс] – Режим доступу : https://web.telegram.org/#/im?p=@overdosehelp_bot – Дата доступу 1.04.2020
3. Инструкция по оказанию первой доврачебной помощи [Електронний ресурс] – Режим доступу : https://www.dzhmao.ru/info/instruktsiya-po-okazaniyu-pervoy-dovrachebnoy-pomoshchi-.php – Дата доступу 3.04.2020
4. Регистрация бота в Telegram – BotFather [Електронний ресурс] – Режим доступу : https://way23.ru/%D1%80%D0%B5%D0%B3%D0%B8%D1%81%D1%82%D1%80%D0%B0%D1%86%D0%B8%D1%8F-%D0%B1%D0%BE%D1%82%D0%B0-%D0%B2-telegram/ - Дата доступу : 10.04.2020
5. Создание бота в Telegram. Основы [Електронний ресурс] – Режим доступу : https://wibe.team/sozdanie-bota-v-telegram/ - Дата доступу : 30.04.2020
6. Rob Miles C# Programming Yellow Book “Cheese” Edition 8.2 November 2016
7. Документація по створенню Telegram ботів [Електронний ресурс] – Режим доступу : https://core.telegram.org/bots - Дата доступу : 5.04.2020

ДОДАТОК

Програмна реалізація

```
namespace Diplom
{
   public static class Program
   {
      private static TelegramBotClient Bot;

      public static async Task Main()
      {

         IConfiguration configuration = new ConfigurationBuilder()
         .SetBasePath(Path.Combine(Directory.GetCurrentDirectory()))
         .AddJsonFile("appsettings.json", optional: true, reloadOnChange:
true)
         .AddEnvironmentVariables()
         .Build();

         var botSettings =
configuration.GetSection("BotSettings").Get<BotSettings>();

         var cts = new CancellationTokenSource();

         try
         {
            Bot = new TelegramBotClient(botSettings.BotToken);

            var me = await Bot.GetMeAsync();
```

```
                Console.Title = me.Username;
                Console.WriteLine("GetMe:" + me.Username);

                Console.WriteLine($"Start listening for @{me.Username}");

                await Bot.ReceiveAsync(new
DefaultUpdateHandler(HandleUpdateAsync, HandleErrorAsync), cts.Token);
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.Message);
                // Send cancellation request to stop bot
                cts.Cancel();
            }
        }

        public static async Task HandleUpdateAsync(Update update,
CancellationToken cancellationToken)
        {
            Task handler;

            switch (update.Type)
            {
                case UpdateType.Message:
                    handler = BotOnMessageReceived(update.Message);
                    break;
                case UpdateType.CallbackQuery:
                    handler =
BotOnCallbackQueryReceived(update.CallbackQuery);
```

```
                break;
            default:
                handler = UnknownUpdateHandlerAsync(update);
                break;
        }
        try
        {
            await handler;
        }
        catch (Exception exception)
        {
            await HandleErrorAsync(exception, cancellationToken);
        }
    }

    private static async Task BotOnMessageReceived(Message message)
    {
        if (message == null || message.Type != MessageType.Text)
            return;
        string name = $"{message.From.FirstName}
{message.From.LastName}";

        Console.WriteLine($"{name} відправив повідомлення
'{message.Text}' ");

        switch (message.Text)
        {

            case "/start":
```

```
                string text =
@"Доброго дня. Вас вітає телеграм бот першої домедичної домомоги.
У разі якщо ви хочете допомогти людині, але ще не викликали швидку допомогу – закрийте телеграмм,
наберіть 103 та після дзвінка поверніться сюди.
Перш за все перевірте чи вам та людині якій ви збираєтесь допомагати нічого не загрожує.
У разі якщо загрози вашому життю неможливо уникнути потрібно піти з місця подій.
Список команд :
/start - початок роботи бота
/danger - меню загроз
/hurt - меню пошкоджень
/help - список команд
/project - інформація про проект
";
                await Bot.SendTextMessageAsync(message.From.Id, text);
                break;
            case "/danger":
                var replyKeyboard = new ReplyKeyboardMarkup(new[]
                {
                    new[]
                    {
                        new KeyboardButton("Обвал бурульок, цеглин тощо.") ,
                        new KeyboardButton("Наїзд автомобілем"),
                    },
                    new[]
                    {
                        new KeyboardButton("Удар елекричним струмом") ,
```

```
                    new KeyboardButton("Напад собак"),
                },
                new[]
                {
                    new KeyboardButton("Напад на вас людиною якій збираєтесь допомогти") ,
                    new KeyboardButton("Загроз не виявлено"),
                },

            }

            );

            await Bot.SendTextMessageAsync(message.From.Id, "Виберіть тип можливої небезпеки", replyMarkup: replyKeyboard);
            replyKeyboard.OneTimeKeyboard = true;
            break;
        case "/hurt":
            var keyboard = new ReplyKeyboardMarkup(new[]
            {
                new[]
                {
                    new KeyboardButton("Поранення") ,
                    new KeyboardButton("Кровотеча"),
                },
                new[]
                {
                    new KeyboardButton("Перелом") ,
                    new KeyboardButton("Перелом голови"),
```

```
                    },
                    new[]
                    {
                        new KeyboardButton("Сонячний удар") ,
                        new KeyboardButton("Обмороження"),
                    },
                    new[]
                    {
                        new KeyboardButton("Утоплення") ,
                        new KeyboardButton("Укуси"),
                    },

                }

                );
                await Bot.SendTextMessageAsync(message.From.Id,
"Виберіть вид ушкодження", replyMarkup: keyboard);
                keyboard.OneTimeKeyboard = true;
                break;
            case "/help":
                string help =
        @"Список команд :
        /start - початок роботи бота
        /danger - меню загроз
        /hurt - меню пошкоджень
        /help - список команд
        /project - інформація про проект
        ";
                await Bot.SendTextMessageAsync(message.From.Id, help);
```

```
                    break;
                case "/project":
                    var inlineKeyboard = new InlineKeyboardMarkup(new[]
                    {
                        new []{
                            InlineKeyboardButton.WithUrl("Зв'язатись з
розробником","https://t.me/panda2804"),
                            InlineKeyboardButton.WithUrl("Джерело медичної
інформації","https://www.dzhmao.ru/info/instruktsiya-po-okazaniyu-pervoy-
dovrachebnoy-pomoshchi-.php"),
                        },
                        new []{
                            InlineKeyboardButton.WithCallbackData("↑
info","callback1"),
                            InlineKeyboardButton.WithCallbackData("↑
info","callback2"),
                        }
                    });
                    await Bot.SendTextMessageAsync(message.From.Id, "Цей
телеграм бот є дипломною роботою. Автор не несе відповідальність за
допомогу яку ви надали після ознайомлення з інформацією.", replyMarkup:
inlineKeyboard);
                    break;
                default:
                    break;
            }
            if (message.Text == ("Обвал бурульок, цеглин тощо."))
            {
                string dang1 =
```

@"У разі загрози обвалу вам потрібно перенести людину в більш безпечне місце.

Якщо ви припускаєте що людина могла зашкодити хребет потрібно спочатку підкласти під спину широку дощечку або аналогічний предмет який дозволить залишити кістки в стані спокою.

Якщо постраждалий має ще якісь видимі оком ушкодження, виберіть вид ушкодження із меню ушкоджень щоб йому допомогти /hurt

";

```
                await Bot.SendTextMessageAsync(message.From.Id, dang1
            );

            }
            if (message.Text == ("Наїзд автомобілем"))
            {
                string dang2 =
```

@"У разі якщо людина, якій ви збираєтесь допомагати лежить на проїжджій частині постарайтесь зупинити машину і попросіть водія вам допомогти. Оцініть ситуацію,

чи доцільно буде переносити потерпілого, чи краще залишити його на дорозі до приїзду швидкої допомоги.

Якщо постраждалий має якісь видимі оком ушкодження, виберіть вид ушкодження із меню ушкоджень щоб йому допомогти /hurt";

```
                await Bot.SendTextMessageAsync(message.From.Id, dang2
            );

            }
            if (message.Text == ("Удар елекричним струмом"))
```

```
            {
                string dang3 =
        @"Якщо поряд із потерпілим розлита вода чи інша рідина яка проводить  струм не підходьте.
        Якщо місце поряд із людиною сухе ви можете підійти і звільнити людину від дії струму. Існує 2 варіанти вирішення проблеми:
        --Взяти суху палицю, дощечку або інший предмет що не проводить струм і звільнити постраждалого від струмовідних частин/загроз
        --Можна відтягнути постраждалого за його одяг якщо він сухий та відстає від тіла
        Відтягніть людину від небезпечної частини щонайменше на 8-10 метрів
        В разі необхідності зробіть штучне дихання і напоїть людину водою та чекайте на приїзд швидкої допомоги.
        https://youtu.be/CHp8MTpkocg
        ";

                await Bot.SendTextMessageAsync(message.From.Id, dang3
            );

            }
            if (message.Text == ("Напад собак"))
            {
                string dang4 =
        @"Якщо на вас напала зграя собак спробуйте знешкодити вожака зграї. Він зазвичай біжить атакувати першим.
        Собаки тварини зграйні, у разі якщо ви влучите чимось важким по вожакові(це може бути цеглина або навіть жіноча сумочка)вся зграя побіжить разом із ним геть.
```

Тепер коли загроза життю не актуальна можна допомогти людині. Виберіть вид ушкодження із меню ушкоджень /hurt ";

```
                await Bot.SendTextMessageAsync(message.From.Id, dang4
            );

            }
            if (message.Text == ("Напад на вас людиною, якій збираєтесь допомогти"))
            {
                string dang5 =
        @"Якщо особа знаходиться в стані алкогольного чи наркотичного сп’яніння та поряд із нею є предмети що можуть загрожувати вашому здоров’ю(пляшка, каміння, шприци, палиця тощо.), перш за все треба прибрати ці предмети з зони доступу людини.
            Тепер коли загроза життю не актуальна можна допомогти людині. Виберіть вид ушкодження із меню ушкоджень /hurt";

                await Bot.SendTextMessageAsync(message.From.Id, dang5
            );

            }
            if (message.Text == ("Загроз не виявлено"))
            {
                string dang6 =
        @"Тепер, коли загроза життю не актуальна можна допомогти людині. Виберіть вид ушкодження із меню ушкоджень /hurt";

                await Bot.SendTextMessageAsync(message.From.Id, dang6
```

```
            );
            }

            if (message.Text == ("Поранення"))
            {
                string hurt1 =
        @"При пораненнях неможна:
        • видаляти із рани залишки предметів, тканини згустків крові тощо.
        • промивати поранення
        • видаляли із рани пісок, землю та інші маленькі предмети, оскільки
самим це зробити неможливо
        При пораненнях перш за все потрібно:
        • продезінфікувати руки
        • очистити місце поруч із раною від бруду та залишків крові
        Якщо під рукою виявився йод та чиста хустинка(або те що може її
замінити) можна зробити перев’язку. Спочатку залийте поранення йодом
щоб пляма від йоду була трохи більша зони ураження та перев’яжіть
тканиною.
        ";

                await Bot.SendTextMessageAsync(message.From.Id, hurt1
            );

            }
            if (message.Text == ("Кровотеча"))
            {
                string hurt2 =
        @"Якщо є підозра на внутрішню кровотечу(блідна шкіра, липкий піт,
часте переривчате дихання та частий пульс слабкого наповнення) потрібно
```

забезпечити постраждалому спокійний стан у положенні лежачи чи напівсидячи.

Якщо кровотеча не сильна то потрібно обробити рану йодом та туго перев’язати

Якщо кровотеча сильна можливі декілька випадків її усунення:

• якщо поранена кінцівка не переламана то можна її зігнути в суглобі вище поранення, в ямку покласти вату і зігнувши до кінця зафіксувати(ременем, хусткою, тканиною тощо)

• накласти джгут. Накладати його потрібно вище рани, ближче до тулуба, в місці накладання джгута потрібно обернути кінцівку м’якою прокладкою(хустка, марля тощо). Попередньо судина яка кровоточить повинна бути притиснута пальцями кістки. Джгут накладено правильно, якщо пульсація судини нижче місця його накладення не визначається, а кінцівка блідне. Його можна накладати розтягуванням (еластичний спеціальний джгут) і закручуванням (краватка, скручена хустка, рушник).

Обов’язково зафіксуйте час накладання джгута, в теплу пору року його можна накладати не більше ніж на 2 години, в холодний час на 1 годину. Також запам’ятайте що сильно затягувати джгут теж не можна, бо тоді можливе ушкодження м’язів, нервових кінцівок тощо.";

```
            await Bot.SendTextMessageAsync(message.From.Id, hurt2
        );

        }
        if (message.Text == ("Перелом"))
        {
            string hurt3 =
```

@"Перш за все при переломах потрібно забезпечити спокій кінцівці що постраждала.

Якщо перелом відкритий потрібно зупинити кровотечу та накласти стерильну пов'язку.

Якщо під рукою виявились дощечки(шматок фанери, палка тощо) можна накласти шину на переламану кінцівку.

При відсутній можливості накладання шини потрібно прибинтувати до тіла(тулуба, здорової кінцівки тощо).

";

```
            await Bot.SendTextMessageAsync(message.From.Id, hurt3
        );

        }
        if (message.Text == ("Перелом голови"))
        {
            string hurt4 =
```

@"Якщо ви помітили ознаки перелому голови(кровотеча із вух та ротової порожнини, втрата свідомості тощо) потрібно

• накласти тугу стерильну пов’язку на рану(якщо вона є)

• покласти потерпілого на спину та зафіксувати голову з двох сторін скрученою у валик тканиною.

• в разі появи блювоти повернути голову набік

• при появі задухи внаслідок западання язика висунути нижню щелепу вперед і підтримувати її в такому положенні

• забезпечити повний спокій.

";

```
            await Bot.SendTextMessageAsync(message.From.Id, hurt4
        );
```

```
            }


            if (message.Text == ("Сонячний удар"))
            {
                string hurt5 =
        @"Перш за все потрібно перенести постраждалого в прохолодне місце та покласти його на спину, якщо є потреба звільнити від одежі що ускладнює дихання. Намочити голову та грудну клітину холодною водою, якщо потерпілий не втратив свідомості напоїти холодною водою.

        Якщо порушено дихання і відсутній пульс, провести штучне дихання і зовнішній масаж серця.

        https://youtu.be/CHp8MTpkocg
        ";

                await Bot.SendTextMessageAsync(message.From.Id, hurt5

             );

            }
            if (message.Text == ("Обмороження"))
            {
                string hurt6 =
        @"При незначному замерзанні негайно розтерти та обігріти охолоджену область для усунення спазму судин (виключивши вірогідність пошкодження шкірного покриву або його поранення).

        При втраті чутливості, побілінні потрібно не допустити швидке нагрівання тих зон які були переохолоджені та забезпечити нерухомість.
        ";
```

```
            await Bot.SendTextMessageAsync(message.From.Id, hurt6

        );

        }
        if (message.Text == ("Утоплення"))
        {
            string hurt7 =
```

@"Людина що надає допомогу повинна не тільки сама добре плавати і пірнати, але і знати прийоми транспортування потерпілого, вміти звільнятися від його захоплень.

Після того як потерпілого дістали із води необхідно очистити рот і глотку (відкрити рот, видалити потрапив пісок, обережно витягнути мову і зафіксувати його до підборіддя бинтом або хусткою, кінці якого зав'язати на потилиці) та видалити воду з дихальних шляхів (потерпілого покласти животом на коліно, голова і ноги звисають вниз; поплескувати по спині).

Якщо після видалення води потерпілий знаходиться в несвідомому стані, відсутній пульс на сонних артеріях, не дихає, почати робити штучне дихання і зовнішній масаж серця. Проводити до повного відновлення дихання або припинити у випадку появи явних ознак смерті, які повинен констатувати лікар.

При відновленні дихання і свідомості укутати, зігріти, напоїти гарячим міцною кавою, чаєм. Забезпечити повний спокій до приїзду лікаря.

```
        https://youtu.be/CHp8MTpkocg";

            await Bot.SendTextMessageAsync(message.From.Id, hurt7

        );
```

```
            }
            if (message.Text == ("Укуси"))
            {


                var bitekeyboard = new ReplyKeyboardMarkup(new[]
                    {
                    new[]
                    {
                        new KeyboardButton("Укуси отруйних змій ") ,
                    },
                    new[]
                    {
                        new KeyboardButton("Укуси тварин") ,
                        new KeyboardButton("Укуси комах"),
                    },
                });
                await Bot.SendTextMessageAsync(message.From.Id, "Виберіть
вид укусу", replyMarkup: bitekeyboard);
                bitekeyboard.OneTimeKeyboard = true;
            }
            if (message.Text == ("Укуси отруйних змій"))
            {
                string hurt8 =
        @"При укусі отруйної змії потрібно як можна швидше відсмоктати
отруту із поранення(для людини яка надає допомогу це безпечно) та
забезпечити спокій людині щоб отрута поширювалась по організму якомога
повільніше.
```

```
Також потрібно гарно напоїти людину водою.";

                await Bot.SendTextMessageAsync(message.From.Id, hurt8);

            }
            if (message.Text == ("Укуси тварин"))
            {
                string hurt9 =
@"Шкіру навколо місця укусу (подряпини) потрібно змастити йодом та накласти стерильну пов'язку;";

                await Bot.SendTextMessageAsync(message.From.Id, hurt9);

            }
            if (message.Text == ("Укуси комах"))
            {
                string hurt10 =
@"Перш за все потрібно видалити жало, покласти на місце набряку «холод» та забезпечити постраждалого великою кількістю води.
При порушенні дихання і зупинці серця робити штучне дихання і зовнішній масаж серця.
https://youtu.be/CHp8MTpkocg
";

                await Bot.SendTextMessageAsync(message.From.Id, hurt10);

            }

        }
```

```
        // Process Inline Keyboard callback data
        private static async Task
BotOnCallbackQueryReceived(CallbackQuery callbackQuery)
        {

            var message = callbackQuery.Message;
            string buttonText = callbackQuery.Data;
            string name = $"{callbackQuery.From.FirstName}
{callbackQuery.From.LastName}";
            Console.WriteLine($" {name} натиснув кнопку {buttonText}");
            if (callbackQuery.Data == "callback1")
            {
                await Bot.AnswerCallbackQueryAsync(callbackQuery.Id,
"Відкриється чат із розробником");
            }
            else
                await Bot.AnswerCallbackQueryAsync(callbackQuery.Id,
"Відкриється сслилка-джерело інформації по наданню домедичної
допомоги");
        }

        private static async Task UnknownUpdateHandlerAsync(Update
update)
        {
            Console.WriteLine($"Unknown update type: {update.Type}");
        }
```

```
        public static async Task HandleErrorAsync(Exception exception,
CancellationToken cancellationToken)
        {
            Console.WriteLine($"Error:\n{exception.Message}");
        }
    }
}
```