

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**До захисту допущено:
Завідувач кафедри**

Сергій СТИРЕНКО

(підпис)

“ ___ ” _____ 2021 р.

Дипломний проєкт

на здобуття ступеня бакалавра

**за освітньо-професійною програмою “Інженерія програмного
забезпечення комп’ютерних систем”**

спеціальності 121 “Інженерія програмного забезпечення”

на тему: Веб-додаток для вивчення ПДР на основі телеграм-бота

Виконала: студентка IV курсу, групи ІІІ-з72
(шифр групи)

Третяк Олена Вячеславівна

(прізвище, ім'я, по батькові)

(підпис)

Керівник асистент, Таран В. І.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант(нормоконтроль) професор, д. т. н. Сімоненко В. П.

(назва розділу)

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент доц. каф. АУТС, к.т.н. Писаренко А.В.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

(підпис)

Київ – 2021 р.

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)

Освітньо-професійна програма

“Інженерія програмного забезпечення комп’ютерних систем” спеціальності

121 “Інженерія програмного забезпечення”

ЗАТВЕРДЖУЮ

Завідувач кафедри

Сергій СТИРЕНКО

(підпис)

“ ___ ” _____ 2021 р.

ЗАВДАННЯ

на бакалаврський дипломний проєкт студента

Третяк Олені Вячеславівні

1. Тема проєкту Веб-додаток для вивчення ПДР на основі телеграм-бота
керівник проєкту Таран Владислав Ігорович, асистент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від _____ 2021 року № _____ .
Термін здачі студентом закінченого проєкту 20 травня 2021 р.
3. Вихідні дані до проєкту технічна документація. технічне завдання.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які розробляються)
РОЗДІЛ 1. Огляд існуючих рішень та літератури 1.1. Загальні поняття чат-бота.
Класифікація 1.2. Огляд існуючих рішень
РОЗДІЛ 2. Аналіз предметної області. Вибір засобів розробки 2.1. Платформи, на
яких розміщують чат-боти 2.2. Предметна область 2.2.1. Використання Python 2.2.2.
Використання C++ 2.2.3. Використання C# 2.2.4. Використання Java 2.3. Робота з
Bot API

РОЗДІЛ 3. Розробка веб-додатку на основі телеграм-бота 3.1. Розробка функціоналу
3.2. Проектування бази даних 3.3. Архітектура програмного забезпечення

РОЗДІЛ 4. Тестування системи

5. Перелік графічного матеріалу (з точним позначенням обов'язкових креслень)
принципова схема, функціональна система, структурна схема системи
6. Консультанта проекту, з вказівкою розділів проекту, які до них вносяться

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормоконтроль	Сімоненко В.П.		

7. Дата видачі завдання 12 грудня 2020 року

Календарний план

№ п/п	Найменування етапів дипломного проекту	Терміни виконання етапів проекту	Примітки
1.	<i>Затвердження теми проекту</i>	<i>10.12.2021-15.12.2021</i>	
2.	<i>Вивчення та аналіз завдання</i>	<i>15.12.2021-15.03.2021</i>	
3.	<i>Розробка архітектури та загальної структури системи</i>	<i>15.03.2021-25.03.2021</i>	
4.	<i>Розробка структур окремих підсистем</i>	<i>25.03.2021-5.04.2021</i>	
5.	<i>Програмна реалізація системи</i>	<i>5.04.2021-15.04.2021</i>	
6.	<i>Оформлення пояснювальної записки</i>	<i>15.04.2021-20.05.2021</i>	
7.	<i>Захист програмного продукту</i>	<i>25.04.2021</i>	
8.	<i>Передзахист</i>	<i>23.05.2021</i>	
9.	<i>Захист</i>	<i>17.06.2021</i>	

Студент-дипломник

(підпис)

Олена ТРЕТЯК

Керівник проекту

(підпис)

Владислав ТАРАН

Анотація

В бакалаврському дипломному проєкті було розроблено веб-додаток для вивчення правил дорожнього руху на основі телеграм-бота. Проєкт містить чотири розділи: в першому зроблено огляд літератури та існуючих рішень за обраною тематикою, в другому проведено аналіз предметної області та обґрунтовано вибір засобів розробки майбутнього додатку, третій описує розробку телеграм-бота, четвертий – його тестування.

Програма дозволяє підготуватися до складання іспитів, якщо користувач має доступ до мережі Інтернет. Програмний продукт був створений на мові Python.

Структура пояснювальної записки роботи. Пояснювальна записка кваліфікаційної роботи складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. Містить 34 малюнка, 7 таблиць. Повний обсяг бакалаврського проєкту становить 57 сторінок.

Annotation

In the bachelor's degree project, a web application was developed to study traffic rules based on a telegram bot. The project contains four sections: the first reviews the literature and existing solutions on selected topics, the second analyzes the subject area and justifies the choice of tools for future application, the third describes the development of a telegram bot, the fourth - it's testing.

The program allows you to prepare for exams if the user has access to the Internet. The software product was created in Python.

The structure of the explanatory note. The explanatory note of the qualification work consists of an introduction, four sections, conclusions, a list of used sources and appendices. Contains 34 figures, 7 tables. The full volume of the bachelor's project is 57 pages.

ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4		Завдання на дипломний проєкт	2	
2	A4	ІАЛЦ.467100.001 ОА	Відомість проєкту	1	
3	A4	ІАЛЦ.467100.002 ТЗ	Технічне завдання	4	
4	A4	ІАЛЦ.467100.003 ПЗ	Пояснювальна записка	57	
5	A4	ІАЛЦ.467100.004 Д1	Принципова схема	1	
6	A4	ІАЛЦ.467100.005 Д2	Функціональна схема	1	
7	A4	ІАЛЦ.467100.006 Д3	Структурна схема системи	1	
8	A4	ІАЛЦ.467100.007 Д4	Текст програми	7	

					ІАЛЦ.467100.001 ОА			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розробив</i>	<i>Третяк О.В.</i>				<i>Веб-додаток для вивчення ПДР на основі телеграм-бота Опис альбому</i>	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Перевірив</i>	<i>Таран В.І.</i>						<i>1</i>	<i>1</i>
<i>Реценз.</i>						<i>НТУУ «КПІ ім. Ігоря Сікорського», ФІОТ</i>		
<i>Н. Контр.</i>	<i>Сімоненко В.П.</i>							
<i>Затв.</i>	<i>Стіренко С.Г.</i>							

ТЕХНІЧНЕ ЗАВДАННЯ
до дипломного проекту
на тему: «Веб-додаток для вивчення
ПДР на основі телеграм-бота»

Київ – 2021 року

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ	3
1.1 Найменування системи.....	3
1.2 Найменування організації – Замовника і Розробника.....	3
1.3 Перелік документів, на основі яких створюється застосування	3
1.4 Планові терміни початку і закінчення роботи зі створення додатку	3
2. ПРИЧИНИ ДЛЯ РОЗРОБКИ	3
3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ.....	3
4. ДЖЕРЕЛА РОЗРОБКИ.....	4
5. ТЕХНІЧНІ ВИМОГИ.....	4
5.1. Вимоги до розробленого продукту	4
5.2. Вимоги до програмного забезпечення.....	5
5.3. Вимоги до надійності системи.....	5
6. ЕТАПИ РОЗРОБКИ	5

					ІАЛЦ.467100.002 ТЗ							
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Технічне завдання			<i>Літера</i>	<i>Аркуш.</i>	<i>Аркушів</i>		
<i>Розроб.</i>		<i>Третяк О.В.</i>								2	5	
<i>Перевір.</i>		<i>Таран В.І.</i>										
<i>Реценз.</i>												
<i>Н. Контр.</i>		<i>Сімоненко В.П.</i>								<i>НТУУ «КПІ ім. Ігоря Сікорського», ФІОТ</i>		
<i>Затверд.</i>		<i>Стіренко С.Г.</i>										

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

1.1 Найменування системи

Технічне завдання поширюється на розробку бота для підготовки до складання іспитів з правил дорожнього руху у межах бакалаврського випускного проєкту.

Повне найменування системи, що розробляється: «Веб-додаток для вивчення правил дорожнього руху на основі телеграм-бота». Скорочене найменування – «Веб-додаток для вивчення ПДР на основі телеграм-бота».

1.2 Найменування організації – Замовника і Розробника

Замовник додатку: кафедра обчислювальної техніки факультету інформатики та обчислювальної техніки. Адреса: 03056, м. Київ, вул. Політехнічна, 41, корпус 18.

Розробник додатку: Третяк О. В., студентка групи ІІІ-з72. Адреса: 03056, м. Київ, вул. академіка Янгеля, 7.

1.3 Перелік документів, на основі яких створюється застосування

Підставою для розробки даного додатку завдання на виконання проєкт бакалавра.

1.4 Планові терміни початку і закінчення роботи зі створення додатку

Датою початку робіт: 11.02.2021 – день видачі завдання на дипломний проєкт.

Дата закінчення робіт: 01.06.2021, дата подання готового дипломного проєкту до захисту.

2. ПРИЧИНИ ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання бакалаврського проєкту за освітньо-професійною програмою “інженерія програмного забезпечення” спеціальності 121 “Інженерія програмного забезпечення”, затверджене кафедрою обчислювальної техніки національного технічного університету України “Київський політехнічний інститут імені Ігоря Сікорського”.

3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою даного проєкту є розробка веб-додатку для допомоги у вивченні правил дорожнього руху. Призначення – надати можливість всім

					<i>ІАЛЦ.467100.002 ТЗ</i>	Арк.
						3
Зм..	Арк.	№ докум.	Підпис	Дата		

користувачам, маючи доступ до Інтернету, тренуватися у проходженні тестів щодо правил дорожнього руху України.

4. ДЖЕРЕЛА РОЗРОБКИ

У якості джерел для розробки беруться: науково-технічна література, публікації в періодичних виданнях, довідники на платформах дистанційного навчання, документація Python, документація Telegram Bot API, публікації в мережі Інтернет з даних питань.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до розробленого продукту

Робота системи Telegram Messenger. У випадку виникнення збоїв на серверах даного продукту, веб-додаток припинить свою роботу до моменту відновлення роботи Telegram Messenger.

Функціональні вимоги до модулів наведені у табл. 5.1.

Таблиця 5.1. Функціональні вимоги до модулів веб-додатку.

Модуль	Функціональна вимога	Пріоритет
Роботи з платформою Telegram Messenger	Система зв'язана ланкою між пристроєм користувача та даними системи	Високий
	Система отримує від сервісу Telegram API дані із запитами користувачів та опрацьовує їх	Високий
	Система використовує дані надані сервісом Telegram API і на основі цього робить запит до модуля роботи з базою даних	Високий
	Система отримує дані від модуля роботи з запитами користувача та опрацьовує їх	Середній
Роботи з базою даних	Система отримує дані від модуля роботи з платформою Telegram messenger platform	Високий
	Система записує результати роботи в базу даних	Високий

5.2. Вимоги до програмного забезпечення

Мінімальні вимоги до технічного забезпечення:

- наявність пристрою з операційною системою Windows/Linux/macOS з можливістю доступу до мережі Інтернет та браузером;
- наявність пристрою з операційною системою Android/iOS з додатком Telegram Messenger.

5.3. Вимоги до надійності системи

Вимоги до апаратної системи: підключення до мережі Інтернет.

При експлуатації веб-додатку можливе виникнення таких аварійних ситуацій:

- відмова систем Telegram Messenger;
- виявлення критичних помилок системи;
- виявлення некритичних помилок системи.

Надійність системи повинна бути забезпечена за рахунок:

- використання системи з операційною системою, вказаною у технічних параметрах;
- своєчасного обслуговування додатків обслуговуючим персоналом;
- швидкого усунення помилок.

Час усунення відмови має бути наступним:

- при збої систем Telegram Messenger – відразу після відновлення;
- при збої критичних помилок – в залежності від помилки, але не більше 3-ох днів.

6. ЕТАПИ РОЗРОБКИ

	Дата
6.1 Вивчення літератури	21.12.2020
6.2 Складання і узгодження технічного завдання	15.02.2021
6.3 Створення модулів системи, що розробляється	15.03.2021
6.4 Створення модулів розробленого курсу	29.03.2021
6.5 Тестування модулів системи	12.04.2021
6.6 Відлагодження і виправлення помилок	26.04.2021
6.7 Оформлення документації дипломного проєкту	20.05.2021

					<i>ІАЛЦ.467100.002 ТЗ</i>	Арк.
						5
Зм..	Арк.	№ докум.	Підпис	Дата		

ПОЯСНЮВАЛЬНА ЗАПИСКА

**до дипломного проекту
на тему: «Веб-додаток для вивчення
ПДР на основі телеграм-бота»**

Київ – 2021 року

ЗМІСТ

Перелік умовних скорочень	3
ВСТУП.....	4
РОЗДІЛ 1. Огляд існуючих рішень та літератури	6
1.1. Загальні поняття чат-бота. Класифікація	6
1.2. Огляд існуючих рішень.....	8
Висновки до розділу 1	14
РОЗДІЛ 2. Аналіз предметної області. Вибір засобів розробки.....	15
2.1. Платформи, на яких розміщують чат-боти.....	15
2.2. Предметна область.....	19
2.2.1. Використання Python.....	20
2.2.2. Використання C++	21
2.2.3. Використання C#.....	23
2.2.4. Використання Java	25
2.3. Робота з Bot API.....	26
Висновки до розділу 2	29
РОЗДІЛ 3. Розробка веб-додатку на основі телеграм-бота.....	30
3.1. Розробка функціоналу	33
3.2. Проектування бази даних	39
3.3. Архітектура програмного забезпечення	43
Висновки до розділу 3	45
РОЗДІЛ 4. Тестування системи.....	46
Висновки до розділу 4	54
Висновки	55
Список використаних джерел.....	57
ДОДАТКИ.....	59
Додаток 1.....	60
Додаток 2.....	62
Додаток 3.....	64
Додаток 4.....	66

					ІАЛЦ.467100.003 ПЗ						
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Пояснювальна записка			<i>Літера</i>	<i>Аркуш.</i>	<i>Аркушів</i>	
<i>Розроб.</i>		<i>Третяк О.В.</i>								2	57
<i>Перевір.</i>		<i>Таран В.І.</i>									
<i>Реценз.</i>											
<i>Н. Контр.</i>		<i>Сімоненко В.П.</i>									
<i>Затверд.</i>		<i>Стіренко С.Г.</i>						<i>НТУУ «КПІ ім. Ігоря Сікорського», ФІОТ</i>			

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API	(Application Programming Interface) – прикладний програмний інтерфейс
ML	(Machine Learning) – машинне навчання
NLP	(Natural-Language Processing) – обробка природної мови
REST	(Representational State Transfer) – передача стану уявлення
БД	База даних
СУБД	(Система управління БД) – комплекс програм, що дозволяють створити БД і маніпулювати даними
ПЗ	Програмне забезпечення

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						3
Зм..	Арк.	№ докум.	Підпис	Дата		

ВСТУП

В сучасних реаліях Інтернет – це універсальне середовище для комунікації, розваг та освіти. На сьогодні в світі існує значна кількість способів, форм і засобів спілкування, і досить немала частка з них якимось чином пов'язана з актуальними сучасними технічними можливостями, які, так чи інакше, представлені з застосуванням можливостей глобальної комп'ютерної мережі. Сьогодні освоюється більше способів для дистанційного навчання.

Інтернет, крім віртуального спілкування, дозволяє скористатися джерелами найрізноманітнішої і корисної для користувачів інформації. Спілкування з товаришами і близькими людьми, контакти зі співробітниками, партнерами, нові знайомства – все це є важливим компонентом повсякденному житті сучасної людини, вибір найбільш підходящих засобів онлайн спілкування у користувача досить великий. На відміну від соціальних мереж, месенджери створені для миттєвого обміну повідомленнями. Пізніше з'явилися можливості створювати канали та розміщувати у них інформацію. А для автоматизації спілкування, допомоги у пошуку інформації, консультаціях у різноманітних питаннях та навіть ігор почали розробляти веб-додатки, якими можна скористуватися, не виходячи з месенджера – так звані, чат-боти – помічники, які спілкуються з користувачами за допомогою повідомлень і мають безліч специфічних функцій. Перевагою є те, що відповідь користувач може отримати майже миттєво, оскільки запити обробляються швидше, ніж це зробить людина.

Актуальність випускної кваліфікаційної роботи обумовлена високою популярністю месенджерів і таких засобів автоматизації як чат-боти серед користувачів мережі Інтернет. Метою дипломного проекту бакалавра є розробка веб-додатку для вивчення правил дорожнього руху на основі бота в месенджері Telegram.

На сьогодні у суспільстві стає поширеною звичкою мати власний транспортний засіб. Чи то буде велосипед для подолання малої відстані щодня,

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						4
Зм..	Арк.	№ докум.	Підпис	Дата		

приміром, доїхати на роботу або з користю для здоров'я покататися з друзями, чи автомобіль для термінових поїздок у відрядження або щоб зменшити переживання за відкладений рейс, необхідністю кудись їхати, купувати квитки (поспіхом шукати їх у мережі). Навіть пішоходи є учасниками дорожнього руху. Але у сучасних реаліях не завжди є можливість ґрунтовно підготуватись до отримання прав, тому певна частина людей починає вивчати правила до моменту вступу в автошколу (курси для водіїв). Так як месенджери мають високу швидкодію, і доступні у будь-який проміжок часу – перерва на роботі, шлях в місце призначення, чекання в черзі, вони є зручними для використання і не потребують додаткового встановлення застосунку на пристрій.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- Аналізувати вибрану предметну область;
- Порівняти аналоги наявних рішень даної задачі;
- Обрати технологію і середовище розробки;
- Розробити веб-додаток на платформі Telegram.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						5
<i>Зм..</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

РОЗДІЛ 1

ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ТА ЛІТЕРАТУРИ

1.1. Загальні поняття чат-бота. Класифікація

Останнім часом інформаційні технології відкривають великі можливості, пов'язані з чат-ботами. Вони настільки міцно увійшли в життя людей, що використовуються в усіх сферах людської діяльності, і з кожним днем їх роль все більше збільшується. В першу чергу це пояснюється тим, що основну частину свого часу люди проводять за смартфоном в e-mail-клієнтах і месенджерах.

Боти – це програми, які виконують різні завдання для користувача, що знаходиться в месенджері. Чат-боти – це спеціально створені акаунти, за якими не закріплена будь-яка людина, а повідомлення, відправлені з них або на них, обробляються зовнішньою системою. Крім того, для користувача спілкування з ботом виглядає як звичайне листування з реальною людиною [1, 2] і має деякі переваги:

- не потрібна установка нових додатків – це зберігає пам'ять пристрою, адже навіть в останніх моделях смартфонів часто доводиться щось видалити для скачування нової програми;
- комунікація відбувається в знайомому інтерфейсі звичного месенджера;
- додатки для спілкування доступні на більшості пристроїв – розробники самі дбають про їх доступність.

Сфери, в яких найчастіше використовують чат-ботів:

- банки: онлайн-розрахунки відсотків за позиками і кредитами;
- оператори зв'язку: технічна підтримка;
- страхування: допомога в заповненні форм і заявок, консультація;
- онлайн-торгівля: консультування з доставки, оплати, адресами точок видачі;
- охорона здоров'я: надання медичних матеріалів, первинна консультація і збір анамнезу;

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						6
Зм..	Арк.	№ докум.	Підпис	Дата		

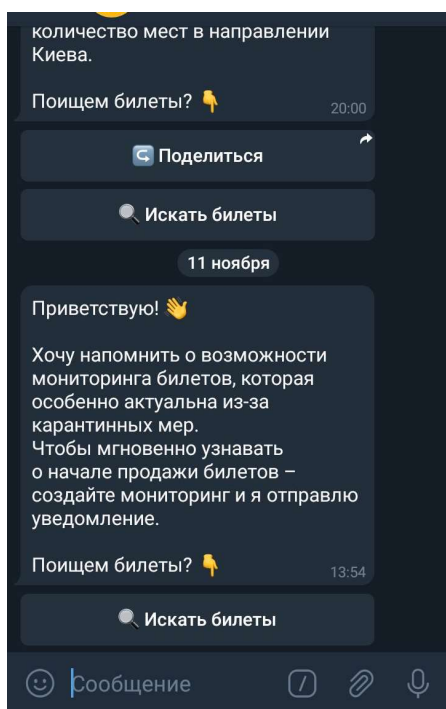
- туризм: агрегатор пропозицій, розсилка гарячих турів, бронювання;
- освіта: розсилки, запрошення на вебінари;
- державні послуги: збір скарг, швидкий доступ до публічних даних;
- HR: підбір підходящих резюме, автоматизація завдань всередині компанії.

За механізмом взаємодії з користувачем існують такі типи чат-ботів:

Кнопковий (рис.1.1а). Найчастіше використовується в месенджерах. Спілкування з клієнтом здійснюється через кнопки з варіантами дій. Віртуальний співрозмовник реагує на них, як на команди, і пропонує користувачеві уточнюючі кнопки або дає відповідь на поставлене запитання;

Текстовий. Найфункціональніший вид бота в месенджерах. Спілкування з ним схоже на комунікацію з людиною, так як робот розпізнає запит, робить аналіз інформації і підбирає для нього найбільш релевантну відповідь із заготовлених;

Вбудований (inline). З'являється всередині діалогу в месенджері (рис.1.1б) після виклику і пропонує варіанти дій. Результатом можна поділитися зі співрозмовником, з яким йшов діалог. Зазвичай використовується для пошуку відповідних локацій, замовлення їжі та інших схожих послуг.



а)



б)

Рис.1.1. Чат-боти на платформі Telegram: а) кнопковий; б) вбудований

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм..	Арк.	№ докум.	Підпис	Дата		7

1.2. Огляд існуючих рішень

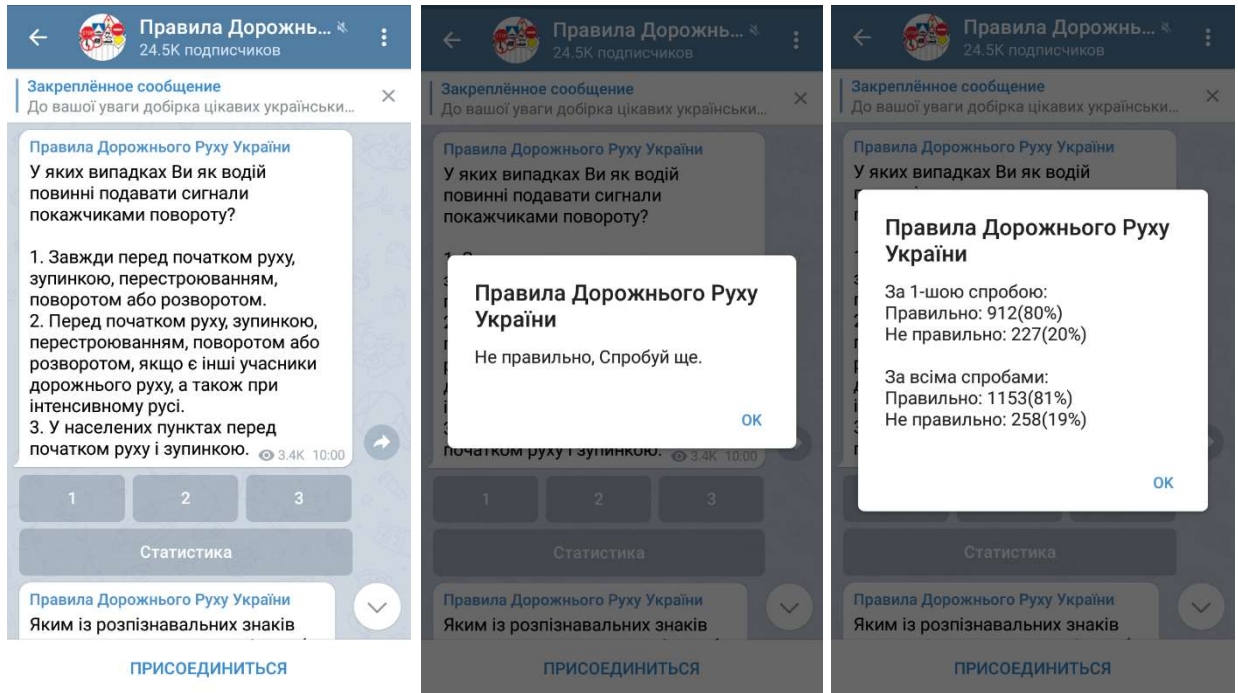
До розробки чат-ботів для залучення користувачів, або ж клієнтів у сфері бізнесу, популярними у застосуванні та користуванні були і залишаються канали у месенджері. Вони не потребують багато знань у створенні. Канал у месенджері – це, як публічна сторінка у соціальних мережах, але з деякими відмінностями:

- так, як канал знаходиться у месенджері, його записи не виділені в окрему стрічку новин, а знаходяться в загальному списку діалогів;
- сповіщення про нові записи надходять у пуш-повідомлення, та їх можна вимкнути.

Мабуть тому, існуючих рішень поставленої задачі більше можна побачити у вигляді каналів.

Канал для вивчення правил дорожнього руху України @PDRUA публікує від одного до трьох повідомлень щодня, кожне з яких є запитанням (іноді з інформацією, яка стосується змін у правилах, новоприйнятих законів) з варіантами відповідей, на яке користувач має змогу дати відповідь і дізнатися, чи коректна вона. Обрати відповідь необхідно, натиснувши на кнопку з відповідною цифрою (рис.1.2а). За правильної відповіді з'являється повідомлення у спливаючому вікні «Правильно», у разі неправильної – повідомлення, показане на рис.1.2б. Натиснувши кнопку «Статистика», можна побачити кількість і відсоткове співвідношення правильно даних відповідей усіх користувачів, що відповіли на дане запитання як за першої спроби, так і за всіма (рис.1.2в).

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						8
Зм..	Арк.	№ докум.	Підпис	Дата		



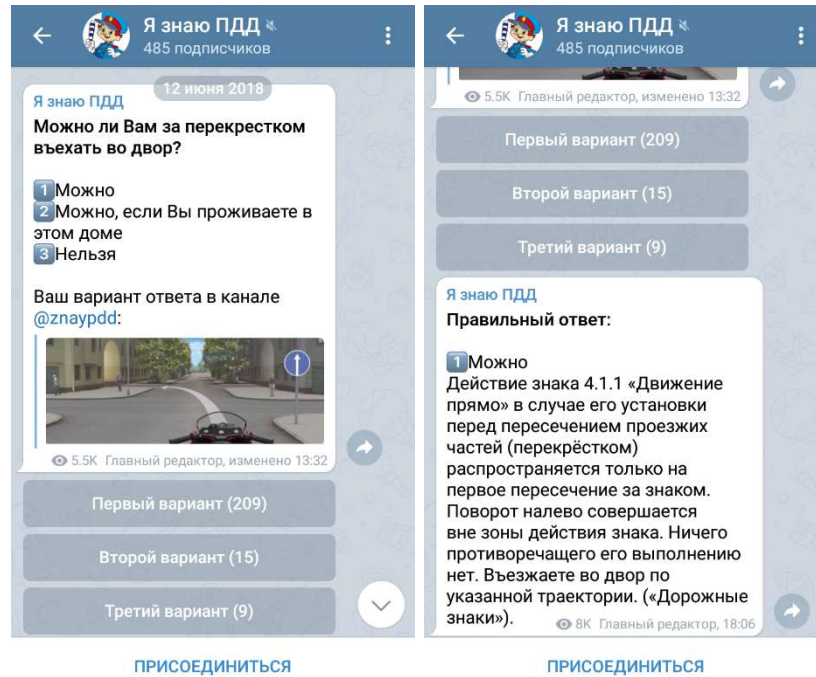
а)

б)

в)

Рис.1.2. Оформлення рішення каналу ПДР України: а) тестове питання; б) повідомлення у разі неправильної відповіді; в) статистика відповіді на питання.

Канал @znaupdd, за описом, публікує запитання протягом дня, на які відповідають користувачі (рис.1.3а), а наприкінці дня публікуються правильні відповіді (рис1.3б). Недоліком є показана кількість відповідей на варіанти питання – якщо кількість значно різниться, то це допомагає здогадатися правильну відповідь, а також показані правильні відповіді у кінці дня, як підказка для тих, хто підписаний на канал не від початку створення і планував попрактикуватися.

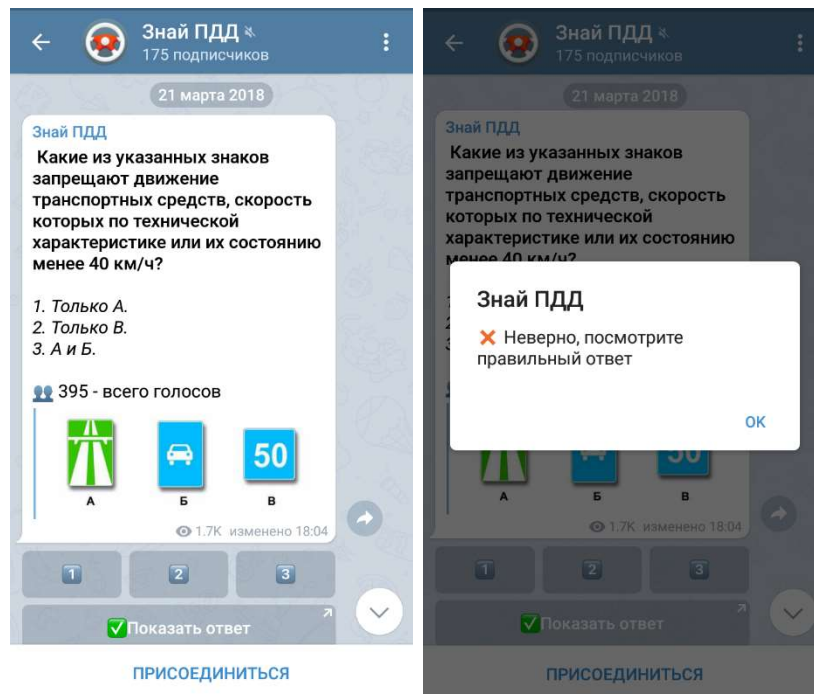


а)

б)

Рис.1.3. Питання про ПДР на каналі: а) питання з варіантами відповідей; б) правильна відповідь після питання.

@znaipdd аналогічно має питання з варіантами відповідей, для відповіді на яке треба натиснути на кнопку (рис.1.4а). За невірної відповіді (рис.1.4б)



а)

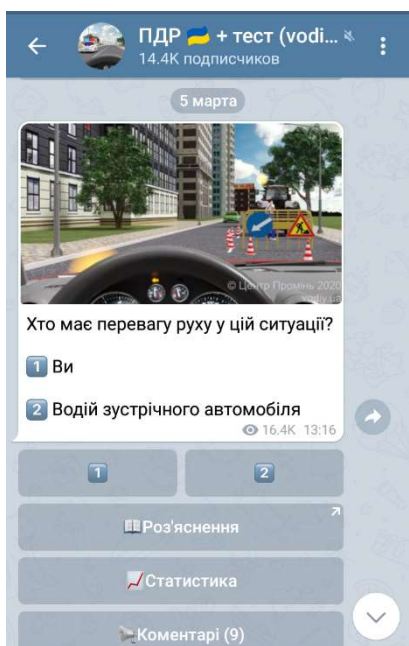
б)

Рис.1.4. Питання про ПДР на каналі: а) питання з варіантами відповідей; б) правильна відповідь після питання.

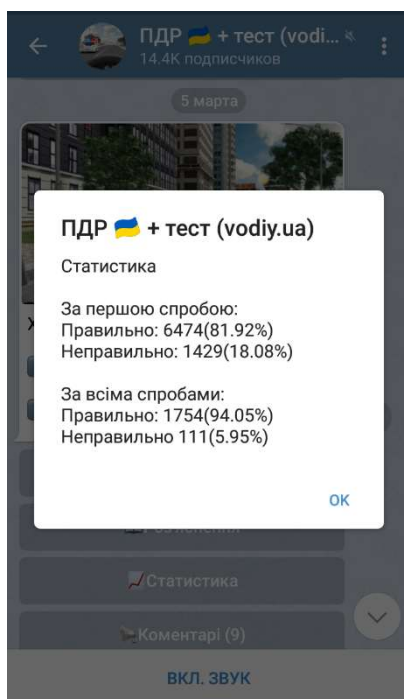
					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
Зм..	Арк.	№ докум.	Підпис	Дата		10

спливаюче повідомлення спонукає користувача одразу подивитися правильну відповідь.

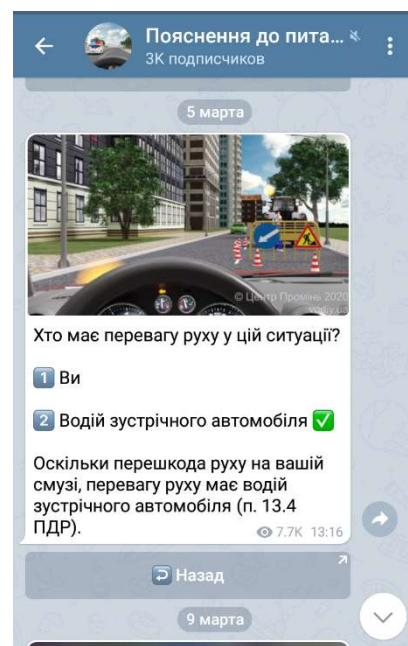
На каналі @vodi_ua також розміщуються питання (рис.1.5а), варіанти відповідей, статистика (рис.1.5б) і можливість коментувати, для цього потрібно натиснути «Коментарі» та підтвердити вхід за допомогою профілю Telegram. Також можна переглянути роз’яснення до питання (рис.1.5в) після наданої відповіді, де також буде показано, яку відповідь обрав користувач і помічено чи коректною вона є. Якщо користувач відповів правильно, про це спливає повідомлення, як показано на рис.1.5г. Якщо відповідь некоректна, виводиться повідомлення про це і повідомляється правильна відповідь (рис.1.5д). В обох випадках виводиться інформація про кількість користувачів, які також вважають вірною дану відповідь та їх відсоткове співвідношення, щодо цього питання.



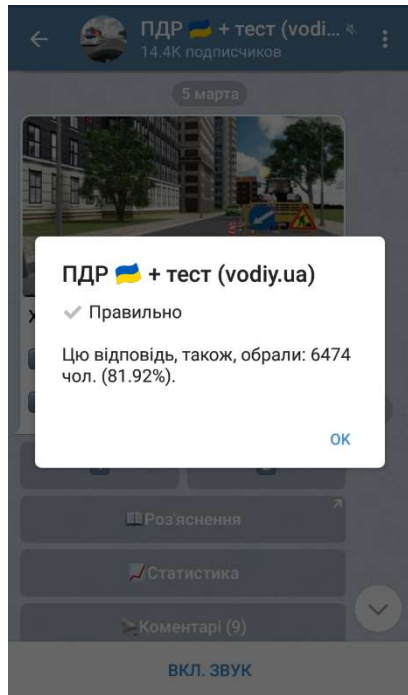
а)



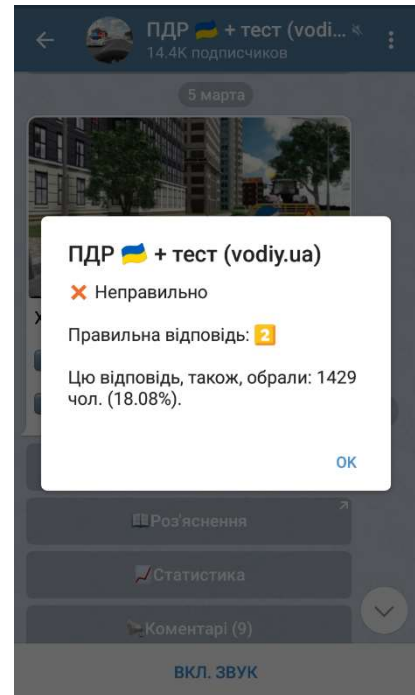
б)



в)



г)



д)

Рис.1.5. Работа канала ПДР + тест: а) ; б) вікно статистики; в) пояснення до питань; г) повідомлення за правильної відповіді; д) повідомлення за невірної відповіді.

Головне меню чат-бота @pdd_testbot [3] представляється кнопками (рис.1.6), при натисканні на які в діалог з ботом відправляються текстові команди, які починаються з символу «/»:

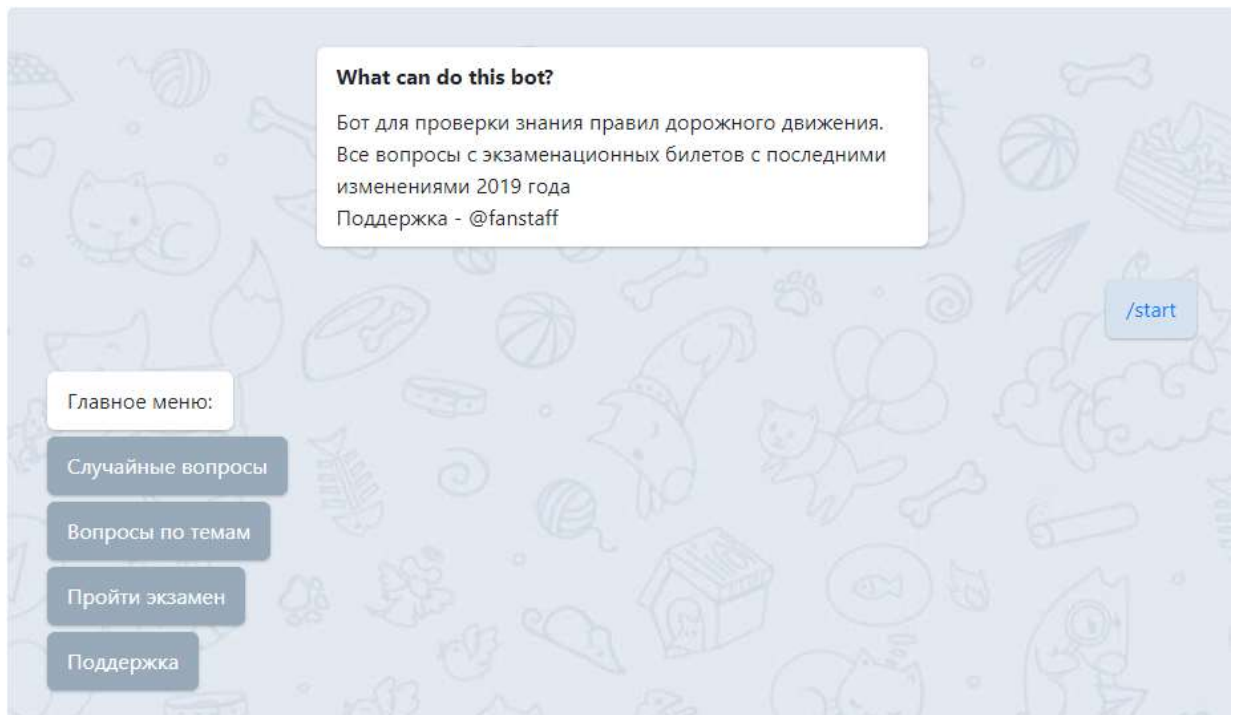


Рис.1.6. Головне меню чат-боту

- / start Головне меню
- / random випадкові питання
- / theme Питання по темам
- / exam пройти іспит
- / stop закінчити іспит
- / help підтримка

Наразі дане рішення не працює – після запуску веб-додатку нічого не відбувається.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
Зм..	Арк.	№ докум.	Підпис	Дата		13

Висновки до розділу 1

З кожним днем месенджери входять у життя все більшої кількості людей. Вони використовуються не тільки для спілкування і обміну файлами, а й для аудіо- та відеодзвінків, ділячи нішу ринку з мобільними операторами. Оскільки, Telegram у сьогоднішні є одним із месенджерів, який має найбільшу аудиторію користувачів, то створення і розміщення веб-додатку в його середовищі є актуальним. У розділі надано поняття «чат-боту», дана класифікація за форматом взаємодії, наведено переваги взаємодії користувача з додатком над взаємодією з оператором. Розглянуто існуючі рішення реалізації веб-додатку на прикладі телеграм-каналів, оскільки рішення у вигляді ботів, які були б у робочому стані після запуску не знайшлося.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						14
<i>Зм..</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

РОЗДІЛ 2

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ. ВИБІР ЗАСОБІВ РОЗРОБКИ

2.1. Платформи, на яких розміщують чат-боти

Бот-платформа – це такий застосунок, на якому за допомогою API можна створити віртуального «штучного» співрозмовника. На сьогоднішній день чат-боти широко поширені у великій кількості популярних месенджерів. Серед них Facebook Messenger, Skype, Telegram, Viber, Whatsapp, і навіть Instagram. Так як месенджери створені на різних платформах, то і API у них різняться. Отже, створити один чат-бот для всіх платформ відразу не можна. Кожне середовище цих платформ має певні характеристики, які відрізняються і відіграють роль при виборі розміщення боту (табл.2.1).

Таблиця 2.1 – Переваги та недоліки найбільш популярних месенджерів

Назва	Переваги	Недоліки
1	2	3
WhatsApp	Доступ по протоколу HTTPS	Суттєве зниження якості медіафайлів, які передаються
	Групові чати до 256 осіб	
	Доступ до чатів в режимі офлайн	Велика затримка у відеочаті
	Зручні голосові і відеодзвінки	
Viber	Push-повідомлення в оффлайн-режимі	Наявність великої кількості реклами
	Вбудовані служби для оплати в мережі Інтернет	Наявність вбудованих покупок
	Наявність публік-акаунтів	
Skype	Демонстрація екрану під час дзвінка або відеотрансляції	Нестабільність з'єднань
	Зручні групові дзвінки і відеоконференції	Сильна залежність від швидкості Інтернет-з'єднання
		Відсутність шифрування

Продовження таблиці 2.1

1	2	3
iMessage	Великий набір стандартних анімацій чату	Велика затримка доставки повідомлень
	Високий ступінь інтеграції зі сторонніми додатками Apple	Працює тільки на системі iOS Неможливість створення чат-ботів
Hangouts	Особливий протокол передачі голосу Google Voice	Не інтуїтивно зрозумілий інтерфейс
	Історія повідомлень зберігається на віддаленому сервері	Надмірне стиснення файлів
	Інтегровані сервіси Google+ і Gmail	
Facebook Messenger	Підтримка ботів	Повідомлення не можна видаляти або редагувати
	Шифрування повідомлень	
	Можливість підтвердження прочитання з Push-повідомлення	Немає підтримки десктопних ОС
	Функція «Секретний чат»	Конфіденційність даних
Самознищення обраних повідомлень через зазначений час		
Telegram	На зручніше платформа для чат-ботів	Поки що відсутні відеодзвінки
	Відкритий вихідний код	
	Вбудований редактор фото	
	Спеціально створений протокол шифрування MTProto	Десктопні версії не підтримують секретні чати
	Секретні чати і повідомлення, що самовидаються	

1. Веб-додатки на платформі Facebook Messenger мають звичний набір функцій. У нього входять:

- текстові повідомлення
- кнопки з варіантами дій;
- до 10 структурних елементів в одному повідомленні;
- рахунки на оплату.

Для того, щоб створити веб-додаток на Фейсбуці, у автора такого чат-боту обов'язково повинна бути публічна сторінка в цій соціальній мережі.

2. Чат-бот в Viber не поступається по функціоналу іншим платформам. Взаємодія з аудиторією ведеться з публічного облікового запису в месенджері. У нього є набір стандартних можливостей і кілька додаткових. А саме:

- розсилки (розсилання повідомлень всім контактам, включаючи тих, хто не підписаний на канал);
- оформлення постів у вигляді «каруселі» з товарами.

3. Ботів в Telegram легко розпізнати по приставці «bot» в назві. Це одна з обов'язкових вимог платформи для власників своїх автоматизованих чатів [4].

Логіка чат-бота в Telegram контролюється за допомогою HTTPS запитів до API платформи. Основні функції, які доступні:

- інтеграція з іншими сервісами (наприклад, «розумним будинком» або роботом-пилососом);
- робота в інлайн-режимі (бот вбудовується в інші діалоги);
- виконання різних завдань (бот може прогнозувати погоду, перекладати тексти і т. д.);
- розважати користувача іграми, інтерактивна взаємодія;
- надсилання текстових повідомлень, коментарів, пошук інформації;
- виконання команди.

Також на платформі є можливості для видозміни інтерфейсу, кастомізації клавіатури, зовнішнього зв'язування. Платформа надає можливість користувачу реалізувати взаємодію з додатком і не вимагає оплати за користування своїми сервісами.

4. На платформі Skype bot нескладні для створення текстові чат-боти, які багато чим схожі на конкурентів. Вони у такий же відповідають на запити користувача за задалегідь прописаним скриптам. У повідомленнях можна додавати і змінювати елементи. З цікавих функцій варто відзначити блок «Меню» з переліком можливостей конкретного чат-бота. Він показується користувачеві при знайомстві з роботом. Велику увагу приділено здатності

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						17
Зм..	Арк.	№ докум.	Підпис	Дата		

бота до саморозвитку. Так, Skype bot використовує Cognitive Service Language API. Це передбачає збір ключових слів з тексту і побудова на них відповідей, облік користувацького досвіду, самостійну генерацію відповідей з напівструктурованого набору шаблонів.

Всі сучасні месенджери мають безліч схожих функцій, таких як відправка повідомлень, медіафайлів, аудіо- та відеосплкування.

На особливу увагу заслуговує месенджер Telegram. На сьогоднішній день цей майданчик найпопулярніший для розміщення чат-ботів. Створити помічника в Telegram просто і зручно. Тут, наприклад, є функціонал для додавання кнопок в діалог. Співрозмовника-робота легко додати до групи або в бесіду, скориставшись відповідною функцією. Боти, як і інші канали, мають ім'я в формі @name, так що їх легко знайти через пошук.

Telegram – це кроссплатформний додаток, розроблений на мові програмування C++, що дозволяє обмінюватися повідомленнями і файлами більшості форматів. Месенджер застосовує спеціально розроблену серверну частину з закритим кодом, що працює на серверах Німеччини і США. Telegram має ряд переваг:

- приватність – всі чати зашифровуються, а повідомлення видаляються через зазначений час;
- швидкість – оперативність надходження повідомлень вище, ніж у аналогів;
- розподіл – сервера Telegram розміщені по всьому світу, що збільшує показник відмовостійкості;
- відсутність підписок і реклами;
- відкритість – використання відкритого протоколу MTProto і API, безкоштовних для всіх;
- відсутність обмежень на розмір повідомлень і максимальний розмір вкладених файлів не перевищує 1.5Гб.

Месенджер Telegram доступний для всіх популярних платформ, наприклад, Android OS, iOS, Windows Phone, Linux та інші.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						18
Зм..	Арк.	№ докум.	Підпис	Дата		

2.2. Предметна область

При пошуку можливих рішень для написання веб-додатку було встановлено, що для розробки бота важливо вміти працювати з REST (Representational State Transfer) API (Application Programming Interface), який надають месенджери, а саме Telegram Bot API.

Слід згадати за протокол передачі гіпертексту (HTTP) – він є одним із відомих протоколів стеку TCP/IP (Transmission Control Protocol / Internet Protocol). Цей протокол розроблявся для отримання і розміщення HTML (HyperText Markup Language) сторінок і зараз використовується для розподілених інформаційних систем. HTTP широко використовується у Інтернет для передачі інформації і представляє найбільш використовуваний прикладний протокол. Визначається подібний протокол типом запит/відповідь. Мається на увазі, що клієнт або веб-браузер, надсилає повідомлення з запитом на сервер, HTTP-протокол сам розпізнає типи повідомлень, які надіслав серверу клієнт способом задання запиту на веб-сторінці, також визначає типи повідомлень які сервер використовуватиме для відповіді. Три найбільш популярні типи повідомлень це: GET, POST і PUT.

HTTP-протокол досить гнучкий, але його не можна назвати безпечним протоколом. Дані відправляються на сервер повідомленням POST у вигляді звичайного тексту, який в свою чергу можуть з легкістю перехопити. Схожа ситуація і з відповідями, що отримуються від сервера – також не шифруються.

Для безпечної комунікації через мережу Інтернет існує безпечний вид HTTP протоколу – HTTPS. За його допомогою можна отримувати доступ або публікувати інформацію на веб-сервері. HTTPS задля захисту від перехоплення даних при передачі їх між клієнтом і сервером, дає змогу використовувати аутентифікацію і шифрування. HTTPS має декілька своїх правил для проходження даних між прикладним і транспортним рівнями.

За технічною класифікацією було визначено, що видів веб-додатків існує три (рис.2.1):

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
Зм..	Арк.	№ докум.	Підпис	Дата		19



Рис.2.1. Технічна класифікація чат-ботів

1. Боти, що самонавчаються (розуміють природні мови) використовують логіку при побудові діалогу або обробку природної мови (NLP) і машинне навчання (ML) для формування відповідей на повідомлення. Або і те, і інше. Користувач може отримати відповідь, навіть, якщо поставив питання нестандартно, бо у такі розробки не закладено попередньо визначений сценарій. Розвиток діалогу, насправді, визначений неявно – на базі, що використовувалася для тренування моделі ML. Тому, щоб послідовність довести до більш логічної, такий тип потребує досить великих наборів даних для тренувань.

2. Скриптові боти (не розуміють природні мови) – стандартні, в яких весь діалог – це заздалегідь сформований шаблон, а «скрипт» являє собою дерево рішень, в якому відповідь на питання відкриває новий, заздалегідь запрограмований сценарій. Діалоги в них зазвичай лінійні і структуровані [2]. Такий бот не зможе відповісти, якщо питання поставлено користувачем у довільній формі. Тому пропонуються варіанти у вигляді вводу з клавіатури відповідного запропонованого варіанта або кнопок.

3. Гібридні представляють собою комбінацію вищеописаних типів і є найбільш вживаними. Діалог з користувачем здійснюється за сценарієм, проте штучний інтелект допомагає розпізнати наміри та виокремити і зберегти певну інформацію (наприклад, ПІБ, дата, місце тощо).

2.2.1. Використання Python

Наостанок, необхідно визначитися для чого використовуватиметься застосунок, так як у бота повинна бути певна задача, тому що інакше він не

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						20
Зм..	Арк.	№ докум.	Підпис	Дата		

представляє ніякої цінності. Відповідно до обраної теми дипломного проекту, призначення веб-додатку полягає у можливості тренування у запам'ятовуванні теорії правил дорожнього руху, в подальшому для складання іспитів. В якості мови, з далі нижче розглянутих, для створення був обраний Python.

Python – це об'єктно-орієнтована мова програмування загального призначення високого рівня з інтегрованою динамічною структурою, спрямована насамперед на підвищення продуктивності самого програміста, ніж коду, який він пише. Часто використовується для підтримки ПЗ у всесвітній павутині та розробки різноманітних додатків. Простіше кажучи, Python дає змогу написати практично що завгодно (веб- / настільні застосунки, ігри, скрипти по автоматизації, комплексні системи розрахунку, системи управління життєзабезпеченням і багато іншого) без відчутних проблем [5]. Більш того, поріг входження низький, а код багато в чому лаконічний і зрозумілий навіть того, хто ніколи на ньому писав. Дана мова дозволяє використовувати пакети та модулі, тобто можна створити свій модуль і в подальшому використовувати його так само, як і стандартні бібліотеки [6]. За рахунок простоти коду, подальший супровід програм, написаних на Python, стає легшим і приємнішим в порівнянні з Java або C++. А з точки зору бізнесу це призводить до скорочення витрат і збільшення продуктивності праці робітника.

Безперечним плюсом є те, що інтерпретатор Python реалізований практично на всіх платформах і операційних системах [7]. Першою такою мовою був Сі, однак його типи даних на відмінних одна від одної машинах могли використовувати різну кількість пам'яті і це служило деякою перепорою при написанні програми, яку дійсно планувалося переносити. Python ж таким недоліком не володіє.

2.2.2. Використання C++

C++ – мова загального призначення середнього рівня, що розроблена як вдосконалення мови С з метою включення об'єктно-орієнтованої парадигми.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						21
Зм..	Арк.	№ докум.	Підпис	Дата		

Структура коду і основний синтаксис цих мов дуже схожі.

Деякі головні моменти у C++ слід відзначити [8, 9]. Тож мова підтримує велику кількість бібліотек, різні типи даних і створювану програму легко структурувати на частини, логічні одиниці. Дана мова є машинно незалежною, проте має залежність від платформи – тобто, скомпільовавши свою розробку на Linux, не вдасться запустити на Windows. Так, як це мова середнього рівня, вона дозволяє розробляти системні програми, наприклад, мережу, драйвери, а також конструювати користувацькі програми такі як відтворювачі музики та редактори зображень.

Доступна бібліотечна підтримка – вбудованих структур даних, алгоритмів, навіть сторонні бібліотеки (для більш оптимізованої, та швидкої зручної розробки програми) і т. д. Мова є процедурною, тому можна відмітити швидкість виконання, але такі дії як динамічне введення тексту, які уповільнюють виконання програми.

C++ підтримує вказівник і безпосередній доступ до пам'яті, що дозволяє користувачу маніпулювати адресою пам'яті. Тобто за допомогою дій з пам'яттю можна програмувати на низькому рівні, контролюючи збереження змінних.

Дану об'єктно-орієнтовану мову вважають однією з найсильніших, що відрізняє її від чистого C. З C++ можна робити реконструйовані та розширені програми – розробляти значні проекти, адже зі збільшенням кількості коду процедурний код стає важчим для підтримки. Також швидкодії сприяє компіляція.

Використовується C++ у найрізноманітніших додатках:

- У відомих операційних системах – Microsoft Windows, Mac OSX, Linux – можна сказати, що ця мова є основною, внаслідок того, що вона швидка, отже підходить для створення операційної системи низького рівня, або ж просто модулів для неї;
- В якості двигунів візуалізації таких веб-браузерів як Chrome і

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
Зм..	Арк.	№ докум.	Підпис	Дата		22

Firefox саме через швидкодiю. Бо користувач не повинен довго чекати завантаження наповнення на свiй екран – i так сталося, що потребуючи малої затримки системи, для розробки використовували C++;

- Беручи до уваги ту саму швидкодiю вiзуалiзацiї, C++ допомiг звести до мiнiмуму затримку у розробцi графiчних та iгрових двигунiв (наприклад, Blender, Photoshop, Unreal-Engine) – для графiчної або цифрової обробки зображень, комп'ютерної графiки. У найпопулярнiших iграх з вiдносно важкою графiкою теж використовують C++ в якостi основної мови програмування. Запропонована швидкодiя середовища сприяє розширенню цiльової аудиторiї, так як навiть на пристроях з малими обчислювальними потужностями оптимiзований застосунок матиме змогу працювати;
- Для баз даних (MongoDB, MySQL, Postgres, Redis та iн.). найпоширенiшими i часто використовуваними є MongoDB та Postgres;
- Для програмування космiчних зондiв, маршрутизаторiв i телефонних комутаторiв – також через швидкiсть дiї;
- Виходячи з того, що C та C++ є мовами нижчого рiвня i бiльш наближенi до апаратних засобiв, iх використовують як компiлятори рiзноманiтних мов програмування.

Зробимо висновок, що розробка програмних засобiв на C++ має переваги у швидкодiї в багатьох сферах застосування та у тому, що вона ближче до апаратних засобiв, нiж той же Python.

2.2.3. Використання C#

C# – проста, сучасна, гнучка та унiверсальна об'єктно-орiєнтована мова програмування. Вона була розроблена у 2001 році компанiєю Microsoft. Застосовується розробниками для створення ПЗ для роботи у теперiшнiх i майбутнiх реалiях завдяки своїй гнучкостi [10].

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						23
Зм..	Арк.	№ докум.	Підпис	Дата		

Характеризується C# наступними показниками:

- Від початку створення мови, закладено мету, щоб вона була не лише легка у вивченні, але щоб могла підтримувати сучасний функціонал у будь-яких видах розробки ПЗ;
- Мова C# була створена як універсальна для забезпечення потреб бізнесу та різних підприємств. Вона також забезпечує потреби в розробці звичайних додатків та веб-додатків, адже можна працювати у напрямі динамічного програмування, асинхронного, обробляти виключення, розширені налагодження та багато чого іншого;
- C# намагається уникати складнощів та неструктурованих особливостей мови. Її синтаксис подібний до Java, C++, Pascal;
- Код є доступним у .NET Foundation, який керується та працює незалежно від Microsoft. Команда, яка працює над вдосконаленням мови робить це активно – додатки, компілятори та інше, пов'язане з цією мовою не важко знайти на Github;
- Мова багатостороння – створені програми .NET можна запускати на платформах Mac, Linux та Windows;
- На C# можна створювати клієнтські програми, компоненти та бібліотеки, сервіси та API, веб-додатки, мобільні додатки, хмарні програми та відеоігри, бо його створено універсальним – для розробки різноманітних програм у різних сферах діяльності [11];
- Зосереджуючись на написанні продуктивного коду, дана мова не дозволяє перетворювати типи, які в подальшому можуть призвести до проблем втрати даних або якихось інших. Тобто, розробники не створять програму, яка в подальшому погано вплине на систему [12];
- Мова розвивається швидше, ніж абиякі інші мови. Остання версія – C# 8.0. Від початку розробки на ній можна було писати клієнтські програми для Windows, та нині доступно написання будь-чого, що

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						24
Зм..	Арк.	№ докум.	Підпис	Дата		

має відношення до консольних програм, хмарних додатків і, звісно, сучасного ПЗ.

Мова дійсно є універсальною та широко використовується у створенні програмних продуктів для різних сфер.

2.2.4. Використання Java

Java – це типізована об'єктно-орієнтована мова програмування, яка розроблена компанією Sun Microsystems. Вона подібна до C і C++, але вважається набагато легшою. Java можна комбінувати з великою кількістю технологій, такими як Android, node js, J2EE тощо. У цілому, ця мова застосовується для створення надійних, масштабованих, портативних та розповсюджених повноцінних додатків. Java також сприяє постійній інтеграції та тестуванню зі застосуванням деяких інструментів.

Java стала однією з найпопулярніших мов програмування у 2020 році. Ця мова багатопотокова і захищена, тож притаманна для написання застосунків у сфері банківських послуг та управління транзакціями. Мобільна операційна система Android працює з API Java. Віднедавна усі великі дані обробляються також з застосуванням функціоналу цієї мови. У дійсності, структура Java разом з іншими структурами, подібними до Spring, використовує надійну комбінацію для сортування залежностей від впровадження та написання серверних додатків у сферах фінансів та ІТ.

Java заманює привабливим набором функцій:

- Самостійна від платформи мова;
- Багатостандартна бібліотека, яка надає можливість легко кодувати. Це допускає створення повністю автономного додатку за допомогою засобів Java;
- Мова володіє високою продуктивністю, оскільки Java підтримує багатопотоковість та паралельність, завдяки чому робить її високоінтерактивною мовою;
- Безпечна і проста.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						25
Зм..	Арк.	№ докум.	Підпис	Дата		

2.3. Робота з Bot API

Bot API – це такий інтерфейс на основі HTTP, який був створений для розробників, що зацікавлені у створенні ботів. Він представляє собою набір способів, за допомогою яких одна комп'ютерна програма може взаємодіяти з іншою. Програмні компоненти взаємодіють між собою за допомогою API, утворюючи ієрархію – високорівневі використовують API низькорівневих.

На початку попереднього пункту згадувались REST – це принципи, які організують взаємодію веб-додатку з сервером за допомогою http-протоколу. Особливість REST полягає в тому, що стан користувача між запитами на запам'ятовується сервером і при кожному запиті обов'язково передаються дані, що ідентифікують користувача, а також параметри, які знадобляться для виконання дії.

Telegram Bot API – інтерфейс на основі http для роботи з ботами на зазначеній платформі. В документації [13] зазначається два способи отримати оновлення: періодичні запити та встановлення веб-хуків (рис.2.2). Оновлення не зберігаються на сервері понад 24 години. Вони повинні бути оброблені протягом цього часу з моменту приходу на сервер. Тому цей спосіб не являється надійним, але є простим – проводиться періодичне опитування сервера Телеграм. Інший спосіб зовсім протилежний – повідомлення від користувача надходить на сервер і месенджер сповіщає про це. Недоліком такого методу є те, що для роботи бота та його встановлення доведеться розгортати повноцінний веб-сервер на тому пристрої, де запускається бот і мати SSL-сертифікат, адже веб-хуки працюватимуть лише на захищеному протоколі HTTPS.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						26
Зм..	Арк.	№ докум.	Підпис	Дата		

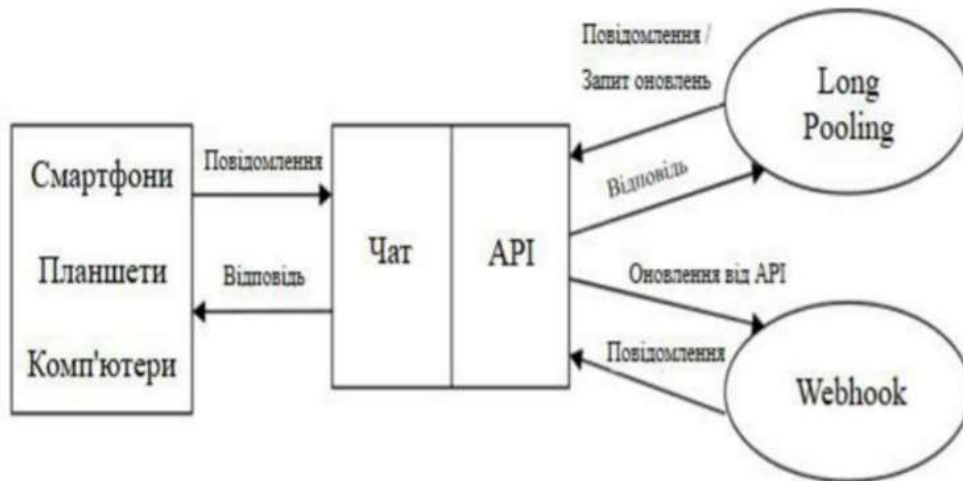


Рис.2.2. Принцип роботи бота у середовищі Telegram

Існують ще такі способи роботи з Bot API:

- AIOGram – це досить простий і повністю асинхронний фреймворк для Telegram Bot API, написаний на Python 3.7 з asyncio і aiohttp. Якщо очікується великий проект з високою відвідуваністю і довгою підтримкою то у цієї бібліотеки немає альтернативи. У бібліотеки є проблеми з документацією, але є чат в самому месенджері, де можна поспілкуватися з розробником даної бібліотеки.
- python-telegram-bot – якщо це щось просте на десяток хендлерів і лінх довго розбиратися то у даної бібліотеки дуже докладні wiki та документація. Можна навіть без досвіду дуже швидко розібратися, як кажуть у обговореннях.
- Пакет python-telegram-bot складається з оболонки API Telegram. Цей інструмент доступний через telegram.Bot-класи. Крім них, є ще модуль telegram.ext, який значно спростить роботу. Модуль telegram.ext містить багато класів, але найважливіші – telegram.ext.Updater і telegram.ext.Dispatcher.Updater відповідає за вибірку нових оновлень від Telegram. В розробці бота імпортуємо все з цього пакету.
- Пакет TeleBot являє собою оболонкою над запитами до TelegramBotAPI. Його використовують для спрощення написаного коду та його мінімізації. Його типи цілком відповідають

визначенням типів API Telegram, крім from поля Message, якому дали назву from_user (так як from – це зарезервований токен Python). Тепер до атрибутів, приміром, message_id можна звернутися напряму, тобто написавши message.message_id. Проте, при створенні програм, які не будуть вимагати високої швидкості виконання, слід зауважити, що атрибут message.chat може належати як конкретному користувачеві, так і чату спільноти, це повністю компенсується перевагами мови.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						28
<i>Зм..</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Висновки до розділу 2

У розділі проведено аналіз платформ, на яких розміщують веб-додатки у вигляді ботів, зокрема, розглянуті Facebook Messenger, Hangouts, iMessage, Skype, Telegram, Viber, WhatsApp. Враховуючи наведені переваги і недоліки, обрано для розміщення додатка Телеграм. Він підтримується на всіх операційних системах для смартфонів та десктопів. Проаналізовано сфери використання деяких мов розробки, та для власного рішення задачі обрано Python, так як це досить гнучка і зручна мова. Проаналізовано роботу з API в зазначеній обраній мові – деякі бібліотеки та методи.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						29
<i>Зм..</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

РОЗДІЛ 3

РОЗРОБКА ВЕБ-ДОДАТКУ НА ОСНОВІ ТЕЛЕГРАМ-БОТА

Бот не має прив'язки до номеру телефону, подібно користувачу. Проте зареєструвати його і отримати ключ доступу Bot API необхідно для подальшого керування веб-додатком [14]. Для цього створено спеціальний бот «BotFather» (рис.3.1). Для спілкування з ним треба надіслати команду «/start».



Рис. 3.1. Іконка офіційного чату з BotFather у Telegram

З'являється повідомлення, показане на рис.3.2. Для утворення облікового запису для нового бота вводимо команду «/newbot», після вказуємо інформацію, якої потребує BotFather – назва та унікальне ім'я, в кінці якого обов'язково має бути записано «bot». Варто пам'ятати, що унікальне ім'я змінити у створеному додатку неможливо – для такої зміни треба буде видалити бота командою /deletebot і створювати нового.

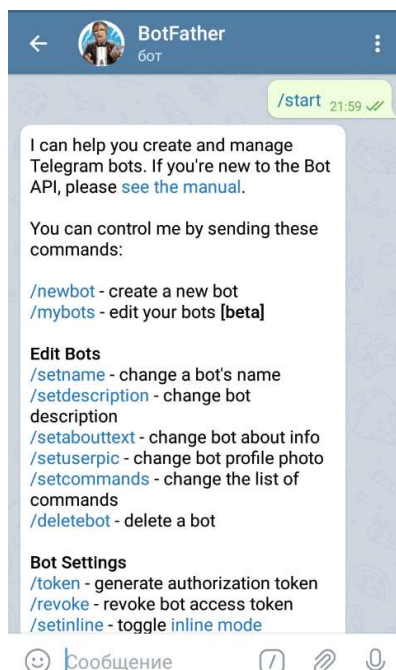


Рис.3.2. Допомога у налаштуванні веб-додатка

						Арк.
						30
Зм..	Арк.	№ докум.	Підпис	Дата	ІАЛЦ.467100.003 ПЗ	

З команд, які надав бот, для налаштування знадобляться представлені в табл. 3.1.

Таблиця 3.1 – Команди для налаштування створеного бота [15]

Команда	Опис
/setname	Змінює існуюче ім'я створеного бота
/token	Повертає токен – ключ доступу Bot API обраного бота
/setdescription	Привласнює текст, який відобразиться при першому відкритті веб-додатку
/setabouttext	Привласнює текст в поле опису бота
/setuserpic	Привласнює обрану картинку на аватар бота
/setcommands	За допомогою команди можна створити список доступних користувачеві команд

За успішної реєстрації «співрозмовник» видає посилання на новоствореного бота (рис.3.3) та повідомляє токен – унікальний id для доступу до HTTP API, завдяки якому вдасться підключитися до боту на сервері.

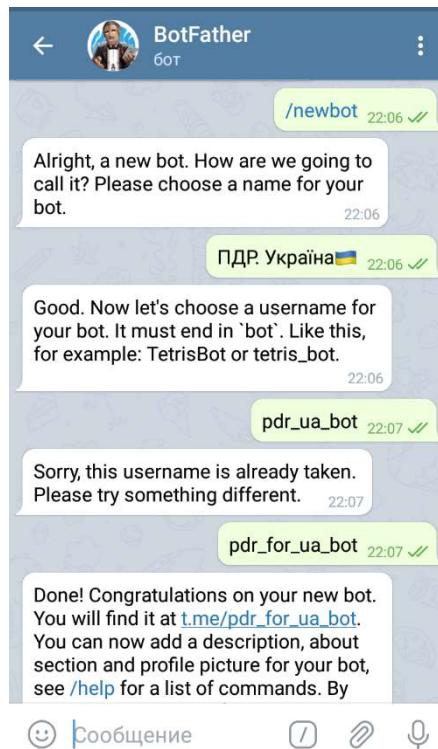


Рис. 3.3. Створення Telegram бота

Тож, коли отримано цей унікальний код, який складається з набору випадкових цифр і букв, копіюємо його в окремий файл для того, щоб програма мала змогу взаємодіяти з конкретним ботом. При заволодінні поточним токеном кимось іншим, за допомогою команди «/revoke» можна його видалити та згенерувати новий.

Тематика веб-додатку відома – він призначений для допомоги майбутнім водіям у вивченні правил дорожнього руху за умови доступу до мережі Інтернет. Для кращого розпізнавання з-поміж інших чатів у месенджері налаштуємо зображення (рис.3.4). За успішного виконання, отримаємо повідомлення «Success! Profile photo updated.».

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
Зм..	Арк.	№ докум.	Підпис	Дата		32



Рис.3.4. Присвоєння зображення

3.1. Розробка функціоналу

Головними вимогами до веб-додатку зазначено наступні:

- При запуску вивести користувачу повідомлення-привітання з метою надання інформації про функціонал;
- На початку роботи бот отримує інформацію про користувача, щоб реагувати на його дії;
- Обравши режим тренування, користувач має отримати випадкове питання з випадкового білета з варіантами відповідей;
- Якщо користувач відповів на питання білету у режимі тренування – надіслати повідомлення про вірну або хибну відповідь з поясненням до питання;
- Якщо користувач отримав повідомлення щодо своєї відповіді у режимі тренування, запропонувати йому тренуватися далі або перейти у початкове меню;
- Обравши режим іспиту, користувач повинен отримати перше питання з випадкового білету, який складається з двадцяти питань;

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						33
Зм..	Арк.	№ докум.	Підпис	Дата		

- У режимі іспиту користувач відповідає на питання одного білету;
- У режимі іспиту підраховується правильна кількість відповідей в межах поточного білету;
- По завершенню тестування у режимі іспиту користувач отримує повідомлення про успішне або неуспішне складання іспиту, а також кількість набраних балів за правильні відповіді;
- По завершенню тестування у режимі іспиту бот пропонує користувачеві повернутися в початкове меню.

Принципова схема рішення щодо зазначених вимог представлена у додатку Д1.

Для роботи з API [16] використовуємо метод sendMessage – він надсилає текстове повідомлення клієнту у месенджер Telegram, маючи наступні параметри:

Таблиця 3.2 – Параметри методу sendMessage [8]

Параметр	Тип	Обов'язковий	Опис
1	2	3	4
chat_id	String або Integer	+	Унікальний ідентифікатор цільового чату або імені цільового каналу.
text	String	+	Текст повідомлення, який відправляється клієнту.
parse_mode	String	–	Надсилається Markdown або HTML, щоб додатки Telegram відображали напівжирний, курсивний, текст з фіксованою шириною або вбудовані URL-адреси в повідомленні бота.

Продовження таблиці 3.2

1	2	3	4
disable_web_page_preview	Boolean	–	Вимикає попередній перегляд посилань в повідомленні.
disable_notification	Boolean	–	Відправляє повідомлення тихо. Користувачі iOS не отримують повідомлення, користувачі Android отримують повідомлення без звуку.
replay_to_message_id	Integer	–	Ідентифікатор вихідного повідомлення.
replay_markup	InlineKeyboardMarkup або ReplyKeyboardMarkup або ReplyKeyboardHide	–	Розширені можливості пошуку інтерфейсу. Являє собою JSON-серіалізований об'єкт для вбудованої клавіатури, яка призначена для користувача.

Оскільки деякі ситуації описувати словесно дуже довго і на це користувач витратить час, більшість питань, у яких можна ілюструвати ситуацію зображенням, матимуть відповідне питанню фото. Для його відправки користувачу необхідно використовувати метод `sendPhoto`, який має параметри, наведені в табл.3.3.

Таблиця 3.3 – Параметри методу `sendPhoto`

Параметр	Тип	Обов'язковий	Опис
1	2	3	4
chat_id	Integer або String	+	Унікальний ідентифікатор цільового чату або імені цільового каналу.

Продовження таблиці 3.3

1	2	3	4
Photo	InputFile або String	+	Надсилає фото. Можна передати file_id як рядок, щоб відправити фотографію яка вже знаходиться на серверах Telegram, або завантажити нову фотографію.
Caption	String	–	Заголовок фотографії, від 0 до 200 символів.
replay_to message_id	Integer	–	Ідентифікатор вихідного повідомлення.

Отримувати інформацію про користувача будемо за допомогою методу User з параметрами (табл.3.4).

Таблиця 3.4 – Параметри методу User

Параметр	Тип	Обов'язковий	Опис
id	Integer	+	Унікальний ідентифікатор користувача або бота
first_name	String	+	Ім'я бота або користувача
last_name	String	–	Прізвище бота або користувача
Username	String	–	Username бота або користувача

На питання білетів давати відповіді буде зручно за допомогою вбудованої клавіатури типу InlineKeyboardButton. Вона з'являється під кожним повідомленням, а отже попереджає необережне випадкове натискання на кнопки клавіатури типу InlineKeyboardMarkup декілька разів підряд на різні питання. Тож, об'єкт InlineKeyboardMarkup має поле inline_keyboard, яке

заповнено масивом у вигляді матриці. Така клавіатура використовуватиметься у режимі тренування та по завершенню тестування у режимі іспиту.

Клавіатура типу InlineKeyboardButton Вона має параметри, наведені в табл.3.5.

Таблиця 3.5 – Параметри об'єкту InlineKeyboardButton [15]

Параметр	Тип	Обов'язковий	Опис
1	2	3	4
Text	String	+	Текст на кнопці
url	String	–	URL, який відкриється при натисканні на кнопку
callback_data	String	–	Дані, які відправлятимуться в callback_query при натисканні на кнопку.
switch_inline_query	String	–	Якщо цей параметр вказаний, то при натисненні на кнопку застосунок запропонує користувачеві обрати будь-який чат, відкриє його і вставити в поле вводу повідомлення юзернейм бота і певний запит для вбудованого режиму. Якщо відправити пусте поле, то буде вставлений лише юзернейм бота.

Продовження таблиці 3.5

1	2	3	4
switch_inline_query_current_chat	String	–	Якщо параметр встановлено, натиснення кнопки введе ім'я бота і вказаний вбудований запит в поле вводу поточного чату. Якщо залишити порожнім, то в цьому випадку в поле буде записано тільки ім'я користувача бота.
callback_game	CallbackGame	–	Опис гри, яка буде запущена, коли користувач натисне кнопку. Примітка: Цей тип кнопки завжди повинен бути першою кнопкою в першому рядку.

Таким чином прийматимемо відповіді, зважаючи на кількість доступних варіантів, які попередньо зазначимо у відповідному масиві, в основній програмі для кожної кнопки прописується (рис.3.5).

```

InlineKeyboardButton(messages.ONE, callback_data='1;' + button_info),
InlineKeyboardButton(messages.TWO, callback_data='2;' + button_info),
InlineKeyboardButton(messages.THREE, callback_data='3;' + button_info),
InlineKeyboardButton(messages.FOUR, callback_data='4;' + button_info)

```

Рис.3.5. Об'єкт InlineKeyboardButton приймає відповідне значення
Самі кнопки описуються у файлі messages.py (рис.3.6).

```

# Кнопки бота
ONE = '1 
TWO = '2 
THREE = '3 
FOUR = '4 

```

Рис.3.6. Визначення кнопок для відповідей у коді

У режимі іспиту якщо обрана користувачем відповідь збігається з номером правильної відповіді у базі, то йому зараховується вірна відповідь шляхом додавання одного балу до загальної кількості правильних відповідей

(рис.3.7) в межах білету, кількість яких на початку тестування рівна нулю, як не складно здогадатися.

```
if int(db.get_write_answer(ticket, question)) == int(user_choice):  
    write_answers += 1
```

Рис.3.7. Зарахування користувачеві правильної відповіді для подальшого визначення результату

Як і при складанні іспиту у відповідній установі, допускається зробити дві помилки для успішного проходження (рис.3.8). Тобто, якщо кількість вірних відповідей складає сімнадцять і менше, користувач не впорався з завданням. Тоді йде звернення системи у файл messages.py до елемента FAILED, повідомляється про неуспішну спробу і пропонується потренуватися у відповідному режимі; за успішної спроби зі змінної PASSED береться повідомлення про вдалу спробу. В обох випадках зазначається кількість правильних відповідей.

```
if 20 - int(write_answers) <= 2:  
    result = messages.PASSED + str(write_answers)  
else:  
    result = messages.FAILED + str(write_answers)
```

Рис.3.8. Код, який відповідає за виведення результату проходження тесту

3.2. Проектування бази даних

SQLAlchemy – це бібліотека на мові Python для роботи з реляційними СУБД з застосуванням технології ORM. Служить для синхронізації об'єктів Python і записів реляційної бази даних (рис.3.9). SQLAlchemy дозволяє описувати структури баз даних і способи взаємодії з ними на мові Python без використання SQL [17].

ORM (Object-Relational Mapping) – це технологія, яка дозволяє зіставляти моделі, типи яких несумісні. Наприклад: таблиця бази даних і об'єкт мови програмування [17].

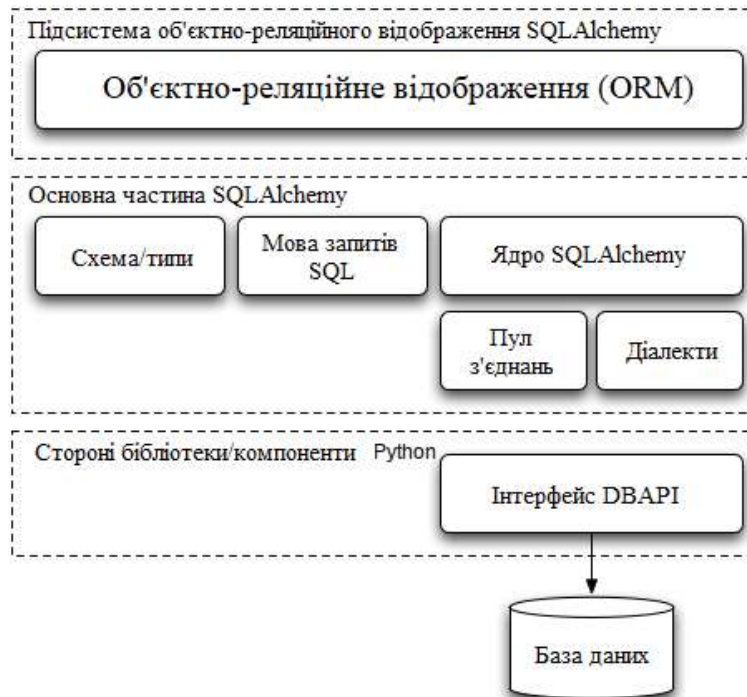


Рис.3.9. Діаграма рівней SQLAlchemy

Переваги використання SQLAlchemy для автоматичної генерації SQL-коду має кілька переваг у порівнянні з ручним написанням SQL:

- **Безпека.** Параметри запитів екрануються, що робить атаки типу впровадження SQL-коду малоімовірними;
- **Продуктивність.** Підвищується імовірність повторного використання запиту до сервера бази даних, що може дозволити йому в деяких випадках застосувати повторно план виконання запиту;
- **Переносимість.** SQLAlchemy, при належному підході, дозволяє писати код на Python, сумісний з декількома back-end СУБД. Незважаючи на стандартизацію мови SQL, між базами даних є відмінності в його реалізації, абстрагуватися від яких і допомагає SQLAlchemy.

В ході створення веб-додатку спроектовано схему взаємодії користувача з базою даних, яка відповідає на запити (рис.3.10).

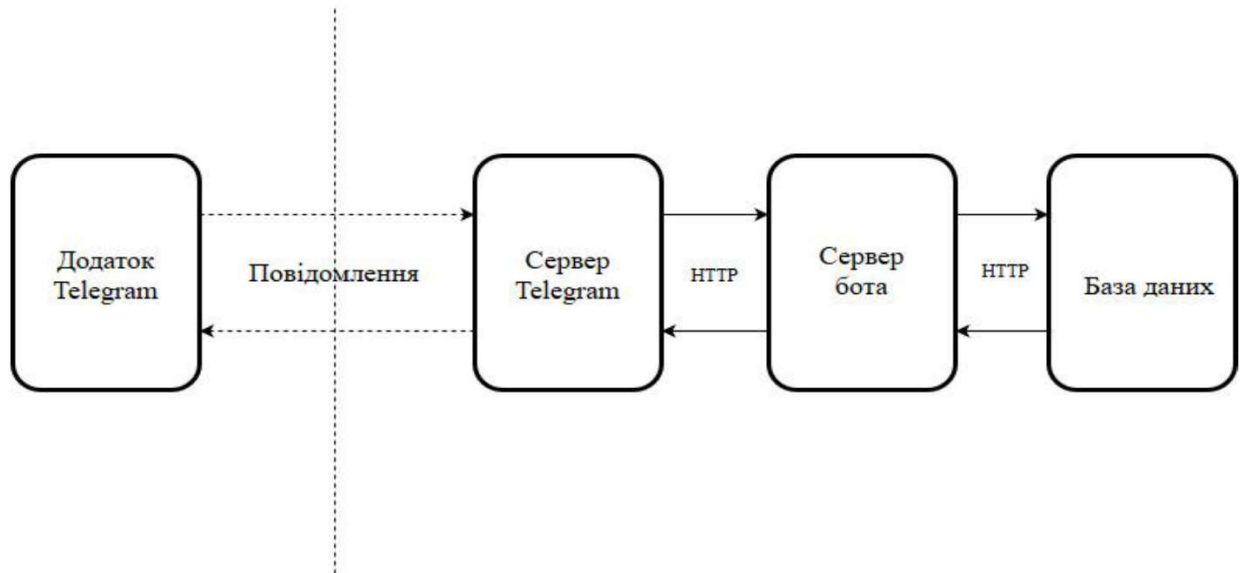


Рис.3.10. Схема співдії користувача з БД

БД записується у файлі в окремі масиви (табл.3.6). Звертання до потрібного масиву здійснюється за допомогою звернення до окремого файлу (рис.3.11), де ці масиви і зберігаються.

Таблиця 3.6. Поля бази даних

Масив	Тип даних	Призначення
questions	srting	Питання до білетів та варіанти відповідей
answers	integer	Номер правильної відповіді
choices	integer	Кількість варіантів відповіді у питанні
comments	srting	Роз'яснення до питання
pictures	srting	Шлях до зображення або вказати, що його немає

```

engine = create_engine('sqlite:///bot.db', echo = True, connect_args={'check_same_thread': False})
Session = sessionmaker()
Base = declarative_base(bind=engine)
  
```

Рис.3.11. Прив'язка до бази даних

Заповнювати базу дуже просто – необхідно зазначити масив даних і перерахувати її складові. Так набір питань білету записується підряд через кому (рис.3.12). Саме ж звернення відбувається через функцію `get_question()`, яка має параметри номеру білета та номеру питання.

```
questions = [
    [
        'Білет 1 - Питання 1\n\
        У яких напрямках Вам дозволено рух в даній ситуації?\n\
        1. Ліворуч і розворот.\n\
        2. Ліворуч.\n\
        3. Прямо, ліворуч і розворот.\n\
        4. Прямо, праворуч, ліворуч і розворот.'
    ],
    'Білет 1 - Питання 2\n\
    Який попереджувальний сигнал вмикається на транспортних
    1. Аварійна сигналізація.\n\
    2. Ближнє світло фар.\n\
    3. Габаритні ліхтарі.'
```

Рис.3.12. Заповнення масиву питань

Ключем є порядковий номер елемента масиву. Наступним заповнюється масив з правильними відповідями (рис.3.13). До його елементів звертаються за допомогою функції `get_write_answer()`, якій передаються ті ж параметри, що і попередній.

```
answers = [
    ['3', '1', '2', '2', '4', '2', '3', '2', '1', '2', '2', '3'],
    ['2', '2', '2', '4', '4', '4', '2', '2', '2', '2', '3', '1']
```

Рис.3.13. Заповнення масиву відповідей

За аналогією заповнюємо базу відповідно до кількості можливих варіантів відповіді у кожному питанні (рис.3.14). Наприклад, можна бачити (див. рис.3.11), що кількість варіантів у першому питанні першого білета складає чотири, другого питання цього ж білету – три. Передаються відповідні параметри у функцію `get_number_of_choices()`, в результаті отримуємо під отриманим повідомленням зазначену кількість кнопок.

```
choices = [
    ['4', '3', '3', '3', '4', '3', '3', '3', '3', '3', '4'],
    ['3', '3', '3', '4', '4', '4', '3', '4', '4', '3', '3']
```

Рис.3.14. Заповнення масиву кількості варіантів відповідей

При заповненні пояснень до питання (рис.3.15) треба врахувати, що апостроф у словах не має позначатися тим самим символом, що і одинарні лапки, інакше система може відмовитися працювати.

```
comments = [  
    'Зелений у вигляді стрілки (стрілок) на чорному фоні дозволяє рух у зазначен  
, 'Аварійна світлова сигналізація повинна бути ввімкнена на всіх механічних тр  
, 'Автомобілі рухаються по дорозі з одностороннім рухом, а дорожній знак 4.3 в  
, 'Знаки «Наближення до залізничного переїзду» встановлюються поза населеним п
```

Рис.3.15. Заповнення масиву роз'яснень до питань

Для зображень створюємо теку в тому ж місці, де знаходиться файл головної програми і даними буде зазначатися шлях до кожного зображення (рис.3.16). Якщо зображення відсутнє у питанні, замість шляху записується "none" і при демонстрації бот не виведе у повідомлення картинку.

```
pictures = [  
    'resur/1-1.jpg'  
, 'resur/1-2.jpg'  
, 'resur/1-3.jpg'  
, 'resur/1-4.jpg'  
, 'none'  
, 'resur/1-6.jpg'  
, 'resur/1-7.jpg'  
, 'resur/1-8.jpg'  
, 'none'  
, 'resur/1-10.jpg'  
, 'resur/1-11.jpg'  
, 'resur/1-12.jpg'  
, 'none'  
, 'none'  
, 'none'  
, 'resur/1-16.jpg'  
, 'none'  
, 'none'  
, 'resur/1-19.jpg'  
, 'none'],
```

Рис.3.16. Заповнення масиву доступу до зображень для питань

До файлу, у якому записані функції запитів імпортується вищезгаданий файл, що містить БД.

3.3. Архітектура програмного забезпечення

Створювана система має архітектуру типу клієнт-сервер. Структурну схему розроблюваної системи представлено на рис.3.10.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
Зм..	Арк.	№ докум.	Підпис	Дата		43

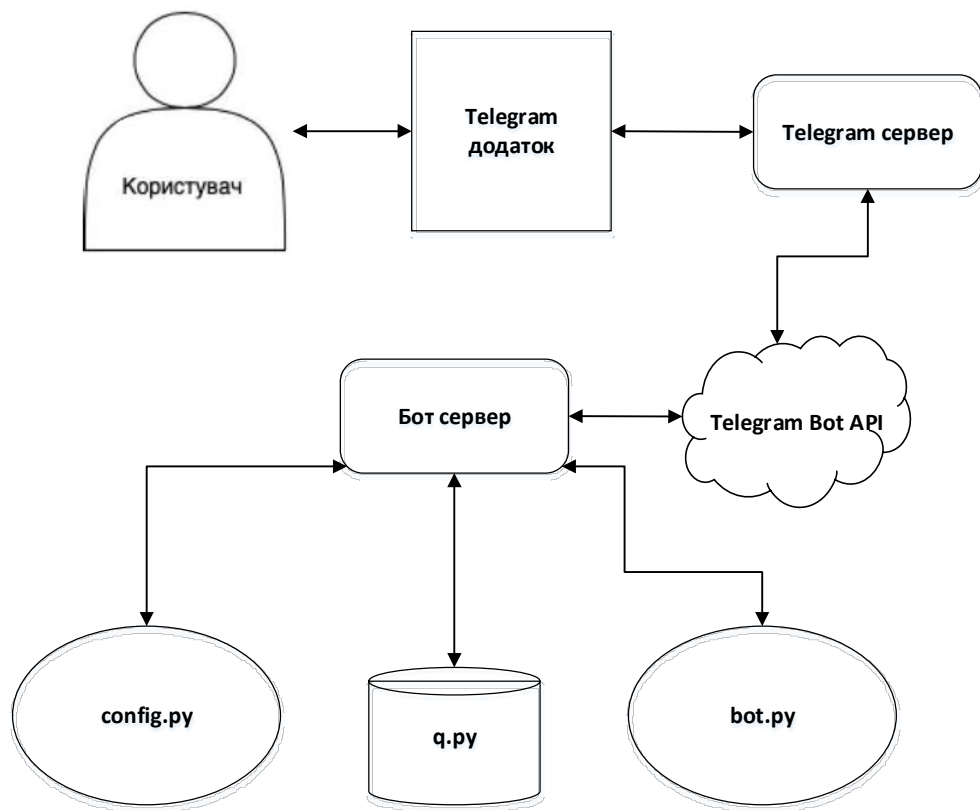


Рис.3.10. Структурна схема розроблюваного веб-додатку

Один з основних елементів системи – сам веб-додаток, який опрацьовує запити від користувача та відправляє їх через API з метою подальшого опрацювання. Телеграм сервер відображає інформацію для збору статистики адміністратором системи. Основний документ, який треба запусити – bot.py – туди підключені майже усі створені файли. Сервером бота на момент тестування слугуватиме ноутбук. У config.py записаний токен розроблюваної системи, за яким користувач взаємодіє з сервером. Як вже описувалося, файл q.py містить базу, дані з якої надходять користувачеві. Переглянути схему також можна у додатку ДЗ.

Висновки до розділу 3

В розділі представлено розробку власного веб-додатку відповідно до функціональних вимог та налаштування його за допомогою BotFather. Для роботи у месенджері отримано токен, завдяки якому система розуміє, з яким ботом їй треба працювати. За допомогою бібліотеки SQLAlchemy, яка не потребує використання самої SQL, налаштована робота з базою даних для бота та розписано, які дані зазначаються. Показана структурна схема взаємодії користувача з сервером через застосунок на платформі Телеграм. В ході розробки усунуто деякі помилки в файлах програми та некоректний вивід інформації з бази. Налаштовано виведення питання і, за наявності, зображення в одному повідомленні. Виконано підрахунок вірних відповідей користувача та на основі отриманого числа вивід повідомлення по завершенню екзамену.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						45
<i>Зм..</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

РОЗДІЛ 4

ТЕСТУВАННЯ СИСТЕМИ

Тестування розробленого програмного забезпечення – важливий етап, так як дає змогу перевірити нижчезазначені цілі:

- Оцінити якість: оцінити швидкодію системи, її безпеку, можливість використовувати на різних платформах та знайти помилки, якщо такі є;
- Перевірити відповідність функціональним вимогам: тут оцінюємо відповідність вимогам системи, виконуваних запланованих дій та їх коректна робота;
- Запобігти дефектам: в подальшому можна покращувати систему, вносячи додаткові функції. В цьому випадку часто змінюється код, може змінюватися і підхід. Тому треба переконатися в тому, що при змінах у будь-якій частині проекту, робота системи не видаватиме помилок.

Для визначення якості продукту за ступенем ізольованості виконуються:

- Модульне тестування – надається оцінка окремим компонентам системи, самостійно один від одного;
- Інтеграційне тестування – перевіряється взаємодія компонентів системи між собою;
- Системне тестування – випробування інтегрованої системи та перевірка її відповідності всім вимогам.

Оскільки вищеописаний веб- додаток створено для месенджера Telegram, для тестування можна використовувати пристрої, що підтримують цей месенджер. Telegram можна використовувати на персональних комп'ютерах, ноутбуках, планшетах чи мобільних пристроях. Також, якщо користувач не бажає встановлювати додаток до себе на пристрій – у Telegram є офіційний сайт web.telegram.org, який має більшість функцій, як і в додатку.

Тестування проводилось за допомогою смартфона Xiaomi Redmi 4A, що має такі характеристики:

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						46
Зм..	Арк.	№ докум.	Підпис	Дата		

- Операційна система: Android 7.1.2;
- Оперативна пам'ять: 2 ГБ;
- Відеоадаптер: Qualcomm Adreno 308;
- Процесор: Qualcomm Snapdragon;
- Вбудована пам'ять: 16 ГБ;
- Telegram v7.7.2

При першій взаємодії користувача з веб-додатком можна бачити привітання, яке створено з допомогою чату з BotFather та команду «Старт», яка запускає застосунок (рис.4.1). Поки команда «Старт» не запущена, користувач може тільки переглянути профіль самого бота (рис.4.2).

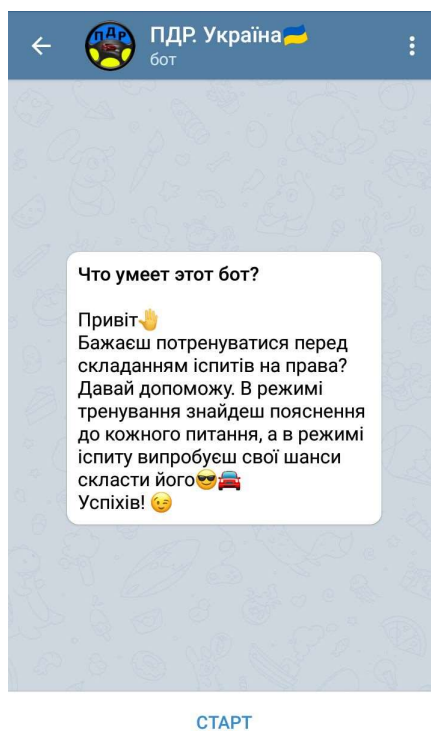


Рис. 4.1. Початок взаємодії користувача з ботом

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
Зм..	Арк.	№ докум.	Підпис	Дата		47

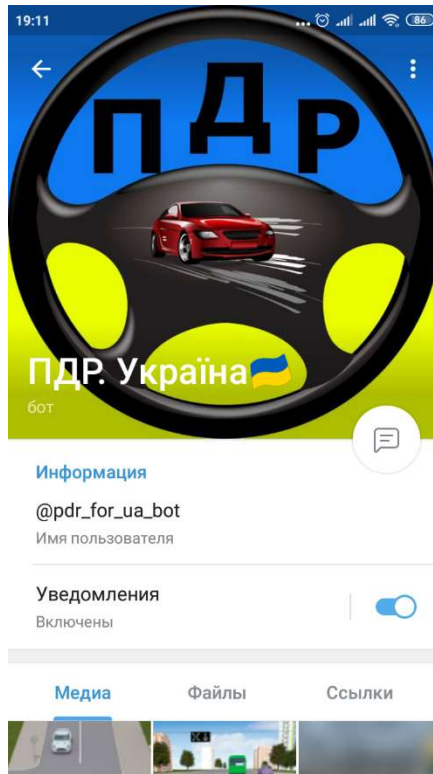


Рис.4.2. Профіль веб-додатку

Після запуску командою «/start» з'являється привітальне повідомлення (рис.4.3), у якому описано команди двох основних режимів – тренування та іспиту.

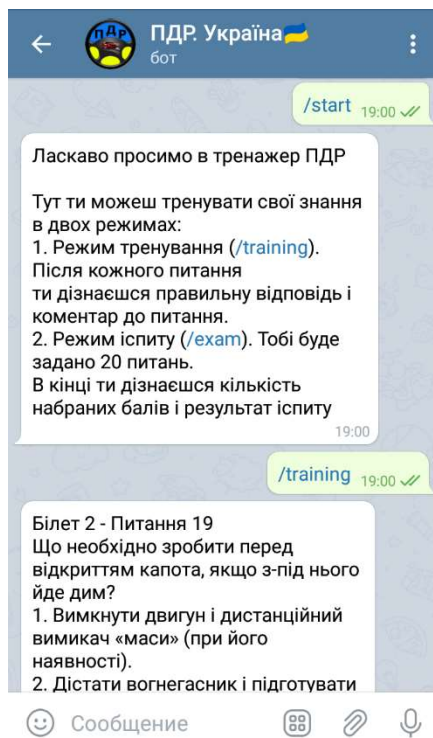
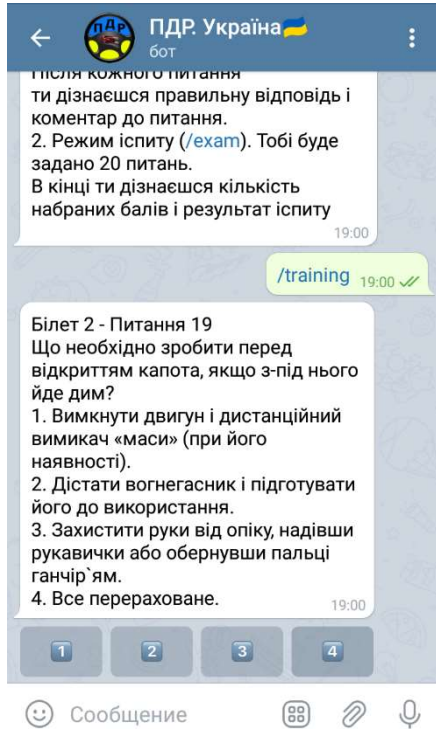


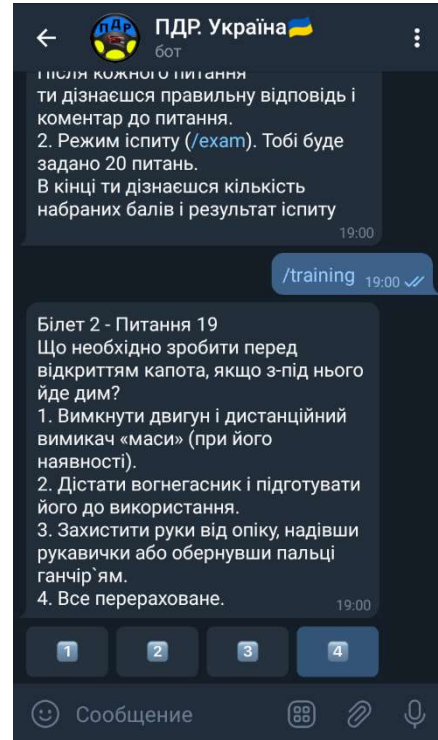
Рис.4.3. Повідомлення-привітання

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
Зм..	Арк.	№ докум.	Підпис	Дата		48

Тож, у користувач має змогу обрати один із режимів. Якщо користувач обрав режим тренування, бот запропонує йому випадкове питання з будь-якого білету (рис.4.4а). Користувач обирає варіант відповіді, який вважає правильним (рис.4.4б).



а)



б)

Рис.4.4. Режим тренування: а) випадкове питання; б) обрання варіанту відповіді користувачем

Якщо користувач відповів правильно, він отримує повідомлення про це і роз'яснення питання (рис.4.5), бо може бути таке, що він не знаючи коректної відповіді, поклався на удачу. Після отримання такого повідомлення, користувач має можливість перейти до наступного питання – так само випадкового з довільного білету, а також натиснувши на «Повернутися в меню», отримає повідомлення для вибору режиму (див. рис.4.3).

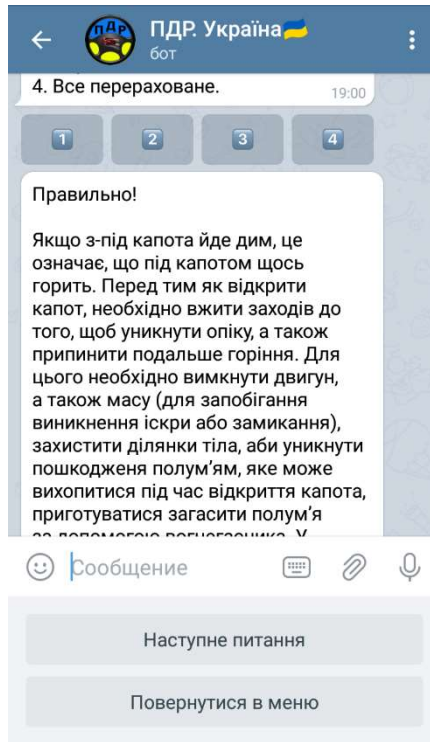


Рис.4.5. Обрано правильний варіант

Якщо користувач відповів некоректно, він отримає таку ж інформацію і можливості, як і за вірної відповіді, але з попередженням, що відповідь неправильна (рис.4.6). За бажанням, клавіатуру можна приховати і викликати натисканням на квадратну іконку з чотирма умовно позначеними кнопками всередині.

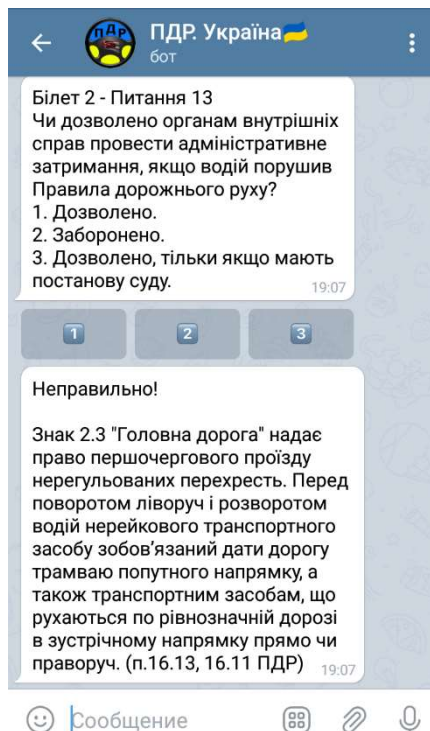


Рис.4.6. Обрано правильний варіант

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						50
Зм..	Арк.	№ докум.	Підпис	Дата		

Режим іспиту передбачає відповіді на двадцять питань одного довільного білету з тих, що прописані у самому додатку (рис.4.7). Питання автоматично задаються користувачеві після його відповіді на поточне.

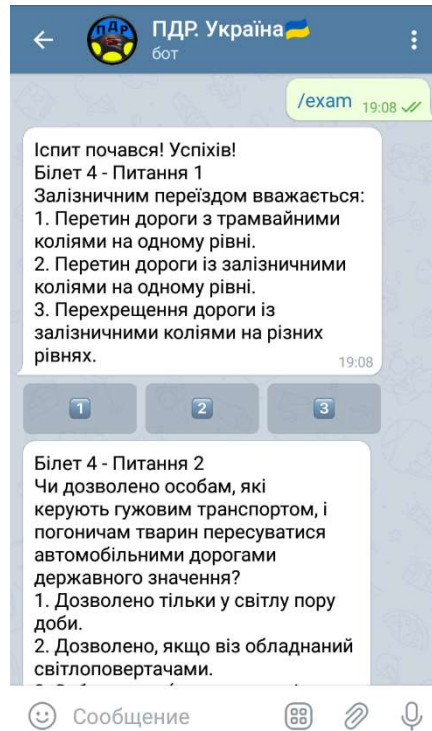


Рис.4.7. Режим іспиту

Під час складання іспиту є можливість відкрити клавіатуру з вже відомими кнопками «Наступне питання» та «Повернутися в меню». При натисканні «Наступне питання» екзамен автоматично припиняється і система переходить у режим тренування і одразу видає випадкове питання з будь-якого білету; при поверненні в меню в режимі іспиту екзамен переривається, система виводить початкове повідомлення-привітання.

Після того, як задано двадцять питань, система, спираючись на кількість вірних відповідей, робить висновок, чи склав користувач іспит – дозволяється зробити дві помилки (рис.4.8), якщо неправильних відповідей три і більше, вважається, що користувач не склав іспит (рис.4.9).

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						51
Зм..	Арк.	№ докум.	Підпис	Дата		

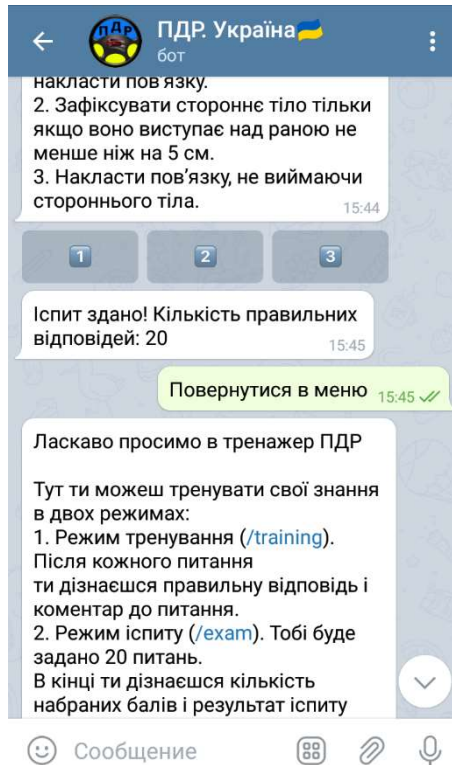


Рис.4.8. Користувач відповів вірно на більше, ніж 17 питань

Після успішних відповідей у режимі іспиту, пропонується повернутися у меню. Після сімнадцяти і менше хибних відповідей користувачеві пропонується потренуватися.

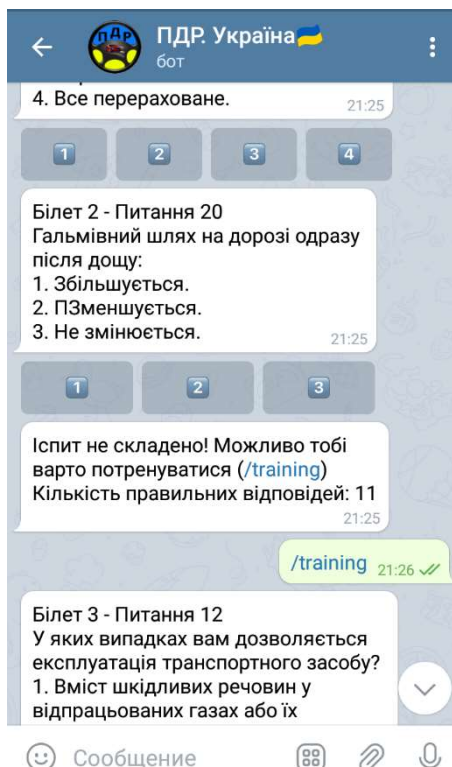


Рис.4.9. Користувач відповів неправильно на більше, ніж 2 питання

Бот приймає лише вищезазначені команди та текст з кнопок. На сторонній текст та/або команди, введені користувачем, він не реагуватиме (рис.4.10).

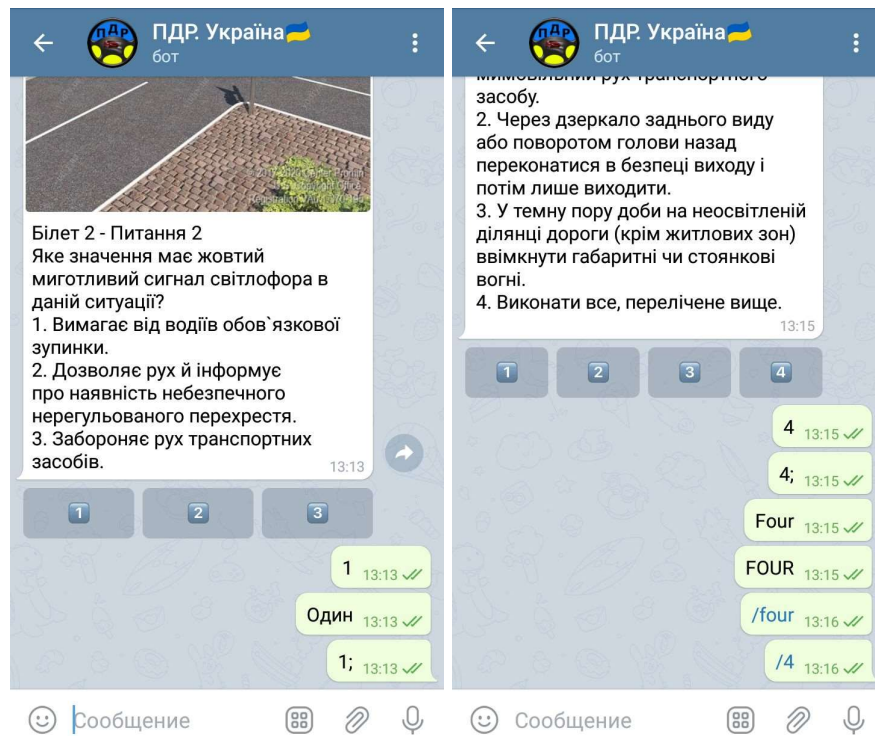


Рис.4.10. Сторонні команди та текст

Висновки до розділу 4

Одним з важливих етапів розробки ПЗ є тестування та відналагодження створеної системи. В ході тестування перевірено відповідність системи встановленим функціональним вимогам, коректну роботу веб-додатку – заплановані дії виконувалися коректно, без збоїв. Серед виявлених і виправлених дефектів були:

- Нумерація одного з зображень до запитання. Система не знайшла картинку, бо її назва відповідала наступному питанню, а наступне питання зображення не передбачало. Проблему було вирішено швидко – змінивши назву. Бот продовжив роботу у звичному режимі.
- Одинарні лапки співпадали з апострофами у словах. Символ апострофу замінений косим апострофом

Критичних помилок, які б впливали на роботу системи, не виявлено.

Отже, розроблена система є надійною та стійкою. Проблеми можуть виникнути тільки за відсутності підключення до Інтернету та/або некоректної роботи серверів месенджеру Telegram. При усуненні цих недоліків веб-додаток може відновитися у роботі.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						54
<i>Зм..</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ВИСНОВКИ

З кожним днем месенджери входять у життя все більшої кількості людей. Вони використовуються не тільки для спілкування і обміну файлами, а й для аудіо- та відеодзвінків, поділяючи нішу ринку з мобільними операторами. Оскільки, Telegram у сьогоднішні є одним із месенджерів, який має найбільшу аудиторію користувачів, то створення і розміщення веб-додатку в його середовищі є актуальним. При роботі над проектом надано поняття «чат-боту», дана класифікація за форматом взаємодії, наведено переваги взаємодії користувача з додатком над співдією з оператором. Розглянуто існуючі рішення реалізації веб-додатку на прикладі телеграм-каналів, оскільки рішення у вигляді ботів, які були б у робочому стані після запуску не знайшлося – тільки опис колись створеного.

Проведено аналіз платформ, на яких розміщують веб-додатки у вигляді ботів, зокрема, розглянуті Facebook Messenger, Hangouts, iMessage, Skype, Telegram, Viber, WhatsApp. Враховуючи наведені переваги і недоліки, обрано для розміщення додатка Телеграм. Він підтримується на всіх операційних системах для смартфонів та десктопів. Проаналізовано сфери використання деяких мов розробки, та для власного рішення задачі обрано Python, так як це досить гнучка і зручна мова. Проаналізовано роботу з API Telegram за допомогою вказаних у другому розділі бібліотек та методів в зазначеній обраній мові.

Створено власний веб-додаток, відповідно до функціональних вимог та налаштування його за допомогою веб-застосунку у Телеграм – BotFather. Для роботи у месенджері отримано токен, завдяки якому система розуміє, з яким ботом їй треба буде працювати. За допомогою бібліотеки SQLAlchemy, яка не потребує використання самої SQL, налаштована робота з базою даних для бота та розписано, які данізначаються. Показана структурна схема взаємодії користувача з сервером через застосунок на платформі Телеграм. В ході розробки усунуто деякі помилки в файлах програми та некоректний вивід інформації з бази. Налаштовано виведення питання і, за наявності, зображення

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						55
<i>Зм..</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

в одному повідомленні. Виконано підрахунок вірних відповідей користувача та на основі отриманого числа вивід повідомлення по завершенню екзамену.

Одним з важливих етапів розробки ПЗ є тестування та відналагодження створеної системи. В ході тестування перевірено відповідність системи встановленим функціональним вимогам, коректну роботу веб-додатку – заплановані дії виконувалися коректно, без збоїв. Серед виявлених і виправлених дефектів були:

- Нумерація одного з зображень до запитання. Система не знайшла картинку, бо її назва відповідала наступному питанню, а наступне питання зображення не передбачало. Проблема було вирішено швидко – змінивши назву. Бот продовжив роботу у звичному режимі.
- Одинарні лапки співпадали з апострофами у словах. Символ апострофу замінений косим апострофом

Критичних помилок, які б впливали на роботу системи, не виявлено. Отже, розроблена система є надійною та стійкою. Проблеми можуть виникнути тільки за відсутності підключення до Інтернету та/або некоректної роботи серверів месенджеру Telegram. При усуненні цих недоліків веб-додаток може відновитися у роботі.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						56
<i>Зм..</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Антонов С. Что такое чат-боты и зачем они нужны? // Inform Бюро [электронный ресурс]. 2018. [Электронный ресурс]. Режим доступа: <https://informburo.kz/cards/chto-takoe-chat-boty-i-zachem-oninuzhny.html/>
2. Технологии создания и применения чат ботов / Новые технологии URL: <https://cyberleninka.ru/article/v/tehnologii-sozdaniya-i-primeneniya-chat-botov>
3. VotoStore. Тесты ПДД [Электронный ресурс]. – Режим доступа: URL: https://ru.botostore.com/c/pdd_testbot/
4. Н. Прохоренок Python 3. Самое необходимое // Н. Прохоренок, В. Дронов – СПб: БХВ-Петербург, 2016. – 461с.
5. Как заработать на создании чат-ботов: опыт белорусского стартапа BotCube / DEVby [Электронный ресурс]]. – Режим доступа: URL: <https://dev.by/lenta/main/kak-zarabotat-na-sozdanii-chat-botov-opyt-belorusskogo-startapa>
6. Марк Лутц, Learning Python :2019, 720с.
7. Python Frequently Asked Questions. [Электронный ресурс]. – Режим доступа: <https://docs.python.org/3/faq/>
8. Б.Страуструп Программирование: принципы и практика использования C++, испр. изд. : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2011. — 1248 с. : ил.
9. Overview of Windows Programming in C++ [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/en-us/cpp/windows/overview-of-windows-programming-in-cpp?view=vs-2019>
10. A tour of the C# language [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>
11. Дж. Албахари C# 5.0. Карманный справочник: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2013. – 288 с.: ил.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
						57
Зм..	Арк.	№ докум.	Підпис	Дата		

12. Э. Тролсен, Ф. Джекис Язык программирования C# 7 и платформы .NET и .NET Core. 8-е издание – М.: ООО «И.Д. Вильямс», 2016. – 1326 с.: ил.
13. Справочник по Bot API. [Электронный ресурс]. – Режим доступа: <https://tigrm.ru/docs/bots/api>
14. Стаття «Створення Telegram Bot» [Електронний ресурс]. – Режим доступу: <https://codeguida.com/post/410>
15. Стаття «Telegram Bots» [Електронний ресурс]. – Режим доступу: <https://core.telegram.org/bots>
16. Telegram Bot API [Електронний ресурс]. – Режим доступу: <https://core.telegram.org/bots/api>
17. Крадущийся тигр, затаившийся SQLAlchemy. Основы. [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/470285/>

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
Зм..	Арк.	№ докум.	Підпис	Дата		58

ДОДАТКИ

ДОДАТОК 1

Веб-додаток для вивчення ПДР на основі телеграм-бота

Принципова схема
ІАЛЦ.467100.004 Д1

Аркушів 1

Київ 2021 р.



ІАЛЦ.467100.004 Д1

Зм.	Арк.	№ докум.	Підпис	Дата
Розробив		Третяк О.В.		
Перевірив		Таран В.І.		
Реценз.				
Н. Контр.		Сімоненко В.П.		
Затв.		Стіренко С.Г.		

Принципова
схема

Літ.	Аркуш	Аркушів
	1	1

НТУУ «КПІ ім. Ігоря Сікорського», ФІОТ

ДОДАТОК 2

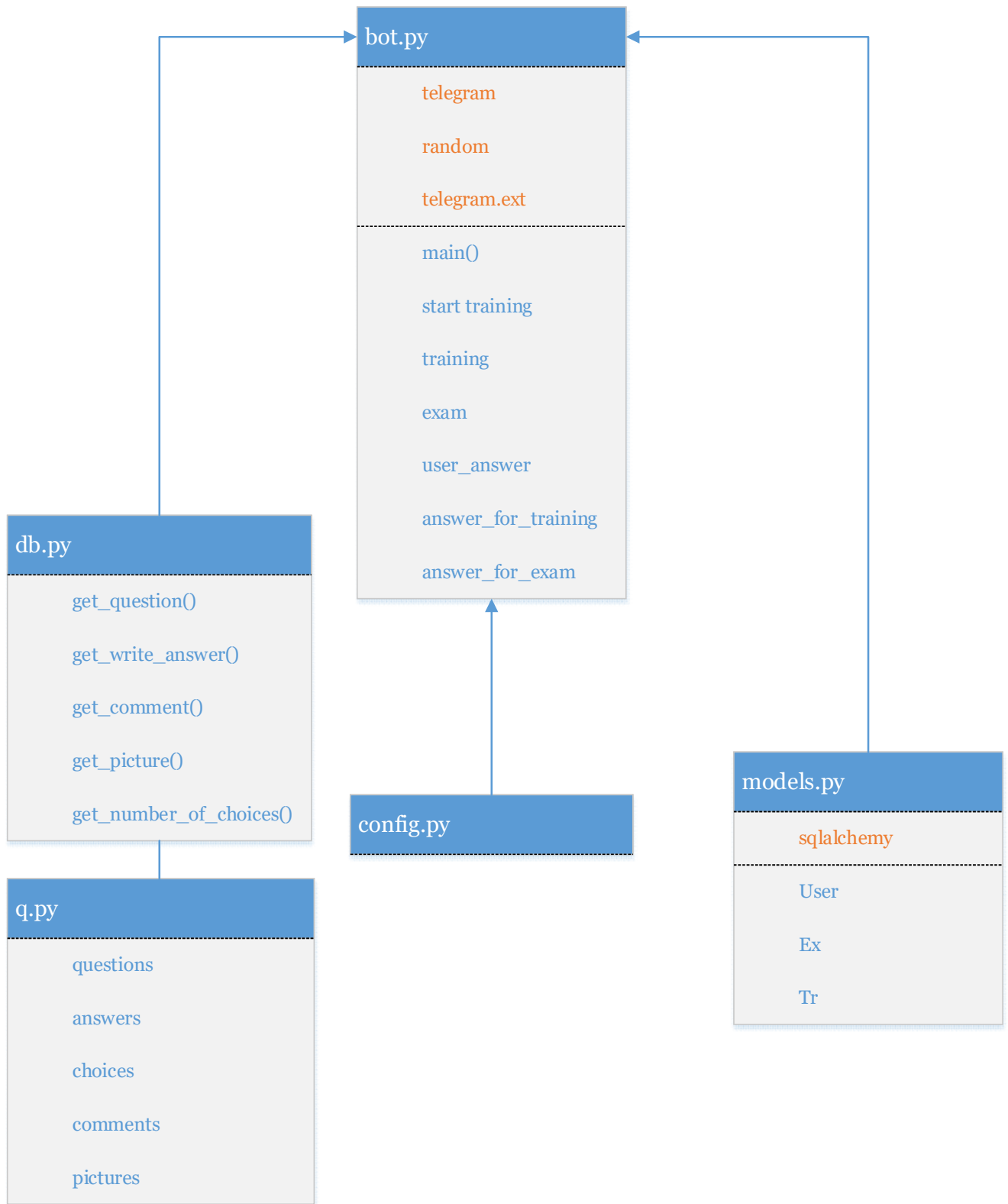
Веб-додаток для вивчення ПДР на основі телеграм-бота

Функціональна схема

ІАЛЦ.467100.005 Д2

Аркушів 1

Київ 2021 р.



					ІАЛЦ.467100.005 Д2		
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
<i>Розробив</i>	<i>Третяк О.В.</i>				<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Перевірив</i>	<i>Таран В.І.</i>					<i>1</i>	<i>1</i>
<i>Реценз.</i>					<i>НТУУ «КПІ ім. Ігоря Сікорського», ФІОТ</i>		
<i>Н. Контр.</i>	<i>Сімоненко В.П.</i>						
<i>Затв.</i>	<i>Стіренко С.Г.</i>						
Функціональна схема							

ДОДАТОК 3

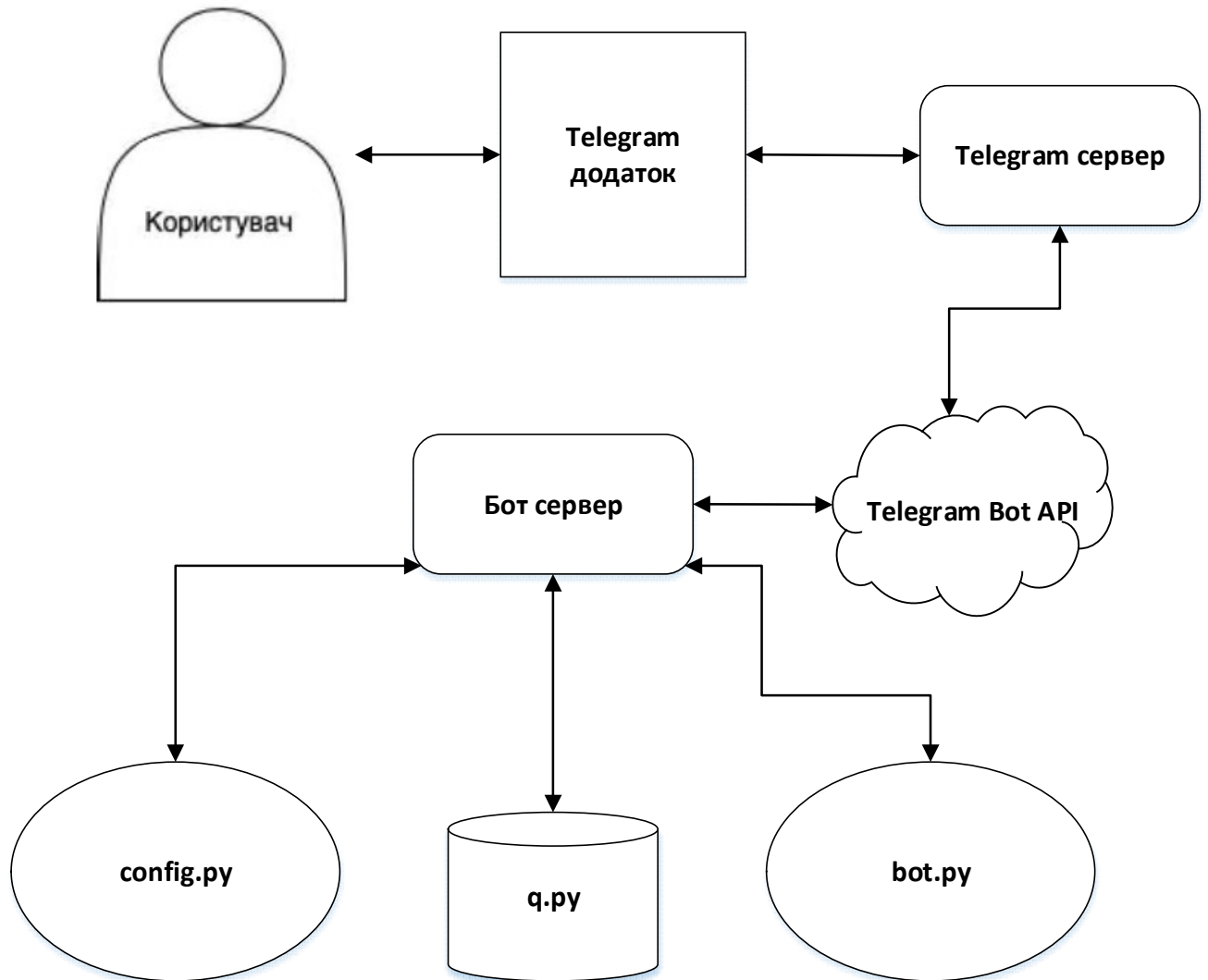
Веб-додаток для вивчення ПДР на основі телеграм-бота

Структурна схема системи

ІАЛЦ.467100.004 ДЗ

Аркушів 1

Київ 2021 р.



ІАЛЦ.467100.006 ДЗ

Зм.	Арк.	№ докум.	Підпис	Дата				
Розробив		Третяк О.В.			Структурна схема системи	Літ.	Аркуш	Аркушів
Перевірів		Таран В.І.					1	1
Реценз.						НТУУ «КПІ ім. Ігоря Сікорського», ФІОТ		
Н. Контр.		Сімоненко В.П.						
Затв.		Стіренко С.Г.						

ДОДАТОК 4

Веб-додаток для вивчення ПДР на основі телеграм-бота

Текст програми

ІАЛЦ.467100.004 Д4

Аркушів 1

Київ 2021 р.

bot.py

```
from telegram.__main__ import main as tmain
from random import randint
from telegram.ext import *
from telegram import *
import db
import messages
import config
from models import User, Ex, Tr, Session, Base, engine

def start(bot, update):
    # Головне меню бота

    bot.send_message(chat_id=update.message.chat_id, text=messages.WELCOME)

    DBSession = Session(bind=engine)
    user_first_name = update.message.from_user.first_name
    user_last_name = update.message.from_user.last_name
    user_name = update.message.from_user.username
    user = update.message.from_user.id
    users = User(id = user, name = user_first_name + ' ' +
user_last_name, user_name = user_name)
    DBSession.add(users)
    DBSession.commit()

def training(bot, update):
    # Метод надсилає випадкове питання з випадкового білету користувачу

    random_ticket = randint(0, 3)
    random_question = randint(0, 19)
    is_exam = 0
    button_info = str(random_ticket) + ';' + str(random_question) + ';' +
str(is_exam)

    three_answers = [[
        InlineKeyboardButton(messages.ONE, callback_data='1;' + button_info),
        InlineKeyboardButton(messages.TWO, callback_data='2;' + button_info),
        InlineKeyboardButton(messages.THREE, callback_data='3;' +
button_info)
    ]]

    four_answers = [[
        InlineKeyboardButton(messages.ONE, callback_data='1;' + button_info),
        InlineKeyboardButton(messages.TWO, callback_data='2;' + button_info),
        InlineKeyboardButton(messages.THREE, callback_data='3;' +
button_info),
        InlineKeyboardButton(messages.FOUR, callback_data='4;' + button_info)
    ]]

    choices = db.get_number_of_choices(random_ticket, random_question)
```

					<i>ІАЛЦ.467100.007 Д4</i>	Арк.
Зм..	Арк.	№ докум.	Підпис	Дата		67

```

    reply_markup = InlineKeyboardMarkup(three_answers) if int(choices) == 3
else InlineKeyboardMarkup(four_answers)

    image = db.get_picture(random_ticket, random_question)
    if image != 'none':
        bot.send_photo(update.message.chat_id, open(image, 'rb'),caption =
db.get_question(random_ticket, random_question),
                    reply_markup=reply_markup)
    else:
        bot.sendMessage(chat_id=update.message.chat_id,
                        text=db.get_question(random_ticket, random_question),
                        reply_markup=reply_markup)

def exam(bot, update):
    # Метод надсилає 20 питань одного білету, а в кінці надсилає результат
іспиту

    random_ticket = randint(0, 3)
    question_number = 0
    write_answers = 0
    is_exam = 1
    button_info = str(random_ticket) + ';' + str(question_number) + ';' +
str(is_exam) + ';' + str(write_answers)

    three_answers = [[
        InlineKeyboardButton(messages.ONE, callback_data='1;' + button_info),
        InlineKeyboardButton(messages.TWO, callback_data='2;' + button_info),
        InlineKeyboardButton(messages.THREE, callback_data='3;' +
button_info)
    ]]

    four_answers = [[
        InlineKeyboardButton(messages.ONE, callback_data='1;' + button_info),
        InlineKeyboardButton(messages.TWO, callback_data='2;' + button_info),
        InlineKeyboardButton(messages.THREE, callback_data='3;' +
button_info),
        InlineKeyboardButton(messages.FOUR, callback_data='4;' +
button_info),
    ]]

    choices = db.get_number_of_choices(random_ticket, question_number)

    reply_markup = InlineKeyboardMarkup(three_answers) if int(choices) == 3
else InlineKeyboardMarkup(four_answers)

    image = db.get_picture(random_ticket, question_number)
    if image != 'none':
        bot.send_photo(update.message.chat_id, open(image, 'rb'),caption =
messages.EXAM_STARTED + '\n' + db.get_question(random_ticket,
question_number),
                    reply_markup=reply_markup)

```

					<i>ІАЛЦ.467100.007 Д4</i>	Арк.
Зм..	Арк.	№ докум.	Підпис	Дата		68

```

else:
    bot.sendMessage(chat_id=update.message.chat_id,
                    text=messages.EXAM_STARTED + '\n' +
db.get_question(random_ticket, question_number),
                    reply_markup=reply_markup)

def user_answer(bot, update):
    # Метод визначає режим роботи бота
    print(update.callback_query.data)
    is_exam = int(update.callback_query.data.split(';')[3])

    if is_exam == 0:
        answer_for_training(bot, update)
    else:
        answer_for_exam(bot, update)

def answer_for_training(bot, update):
    # Метод надсилає користувачеві випадкове питання з випадкового білету

    keyboard = [[str(messages.NEXT)], [str(messages.MENU)]]

    query = update.callback_query
    user_choice = int(query.data.split(';')[0])
    ticket = int(query.data.split(';')[1])
    question = int(query.data.split(';')[2])
    is_exam = int(update.callback_query.data.split(';')[3])

    comment = db.get_comment(ticket, question)
    write_choice = int(db.get_write_answer(ticket, question))

    if user_choice == 5:
        bot.sendMessage(chat_id=query.message.chat_id,
                        text=db.get_question(ticket, question),
                        reply_markup=ReplyKeyboardMarkup(keyboard,
one_time_keyboard=True, resize_keyboard=True))

        image = db.get_picture(ticket, question)
        if image != 'none':
            bot.send_photo(query.message.chat_id, open(image, 'rb'))

    elif user_choice == 6:
        bot.sendMessage(chat_id=query.message.chat.id, text=messages.WELCOME)
        if user_choice == write_choice:
            bot.sendMessage(chat_id=query.message.chat.id, text='Правильно! \n\n'
+ comment,
                            reply_markup=ReplyKeyboardMarkup(keyboard,
one_time_keyboard=True, resize_keyboard=True))#reply_markup=reply_markup)
        else:
            bot.sendMessage(chat_id=query.message.chat.id, text='Неправильно!

```

					<i>ІАЛЦ.467100.007 Д4</i>	Арк.
Зм..	Арк.	№ докум.	Підпис	Дата		69

```

\n\n' + comment,
                                reply_markup=ReplyKeyboardMarkup(keyboard,
one_time_keyboard=True, resize_keyboard=True))#reply_markup= keyboard)

DB2Session = Session(bind = engine)
a = 1 if user_choice == write_choice else 0
t_mis = Tr(id = query.message.chat.id , mis = a, ticket = ticket + 1)
DB2Session.add(t_mis)
DB2Session.commit()

def answer_for_exam(bot, update):
    # Метод надсилає 20 питань користувачу, а в кінці підраховує кількість
    # правильних відповідей

    query = update.callback_query
    user_choice = int(query.data.split(';')[0])
    ticket = int(query.data.split(';')[1])
    question = int(query.data.split(';')[2])
    is_exam = int(update.callback_query.data.split(';')[3])
    write_answers = int(query.data.split(';')[4])

    if int(db.get_write_answer(ticket, question)) == int(user_choice):
        write_answers += 1

    question += 1

    button_info = str(ticket) + ';' + str(question) + ';' + str(is_exam) +
';' + str(write_answers)
    print(button_info)

    if question < 20:
        three_answers = [[
            InlineKeyboardButton(messages.ONE, callback_data='1;' +
button_info),
            InlineKeyboardButton(messages.TWO, callback_data='2;' +
button_info),
            InlineKeyboardButton(messages.THREE, callback_data='3;' +
button_info)
        ]]

        four_answers = [[
            InlineKeyboardButton(messages.ONE, callback_data='1;' +
button_info),
            InlineKeyboardButton(messages.TWO, callback_data='2;' +
button_info),
            InlineKeyboardButton(messages.THREE, callback_data='3;' +
button_info),
            InlineKeyboardButton(messages.FOUR, callback_data='4;' +
button_info),
        ]]
        choices = db.get_number_of_choices(ticket, question)

```

					<i>ІАЛЦ.467100.007 Д4</i>	Арк.
						70
Зм..	Арк.	№ докум.	Підпис	Дата		

```

    if int(choices) == 3:
        keyboard = InlineKeyboardMarkup(three_answers)
    else:
        keyboard = InlineKeyboardMarkup(four_answers)

    image = db.get_picture(ticket, question)
    if image != 'none':
        bot.send_photo(chat_id=query.message.chat_id, photo=open(image,
'rb'),caption =db.get_question(ticket, question), reply_markup=keyboard)
    else:
        bot.sendMessage(chat_id=query.message.chat.id,
                        text=db.get_question(ticket, question),
reply_markup=keyboard)

else:
    keyboard = [[str(messages.MENU)]]

    if 20 - int(write_answers) <= 2:
        result = messages.PASSED + str(write_answers)
    else:
        result = messages.FAILED + str(write_answers)

    bot.sendMessage(chat_id = query.message.chat.id,
                    text = result,
                    reply_markup=ReplyKeyboardMarkup(keyboard,
one_time_keyboard=True, resize_keyboard=True))
    DB3Session = Session(bind = engine)
    mis = Ex(id = query.message.chat.id , k_mis = (20 -
int(write_answers)), ticket = ticket + 1)
    DB3Session.add(mis)
    DB3Session.commit()

def main():
    updater = Updater(config.TOKEN, workers=3, use_context=False)
    dp = updater.dispatcher
    dp.add_handler(CommandHandler("start", start, run_async=True))
    dp.add_handler(CommandHandler("training", training, run_async=True))
    dp.add_handler(CommandHandler("exam", exam, run_async=True))
    dp.add_handler(RegexHandler('^(Повернутися в меню)$', start,
run_async=True))
    dp.add_handler(RegexHandler('^(Наступне питання□)$', training,
run_async=True))
    dp.add_handler(RegexHandler('^(Перескласти іспит□)$', exam,
run_async=True))
    dp.add_handler(CallbackQueryHandler(user_answer, run_async=True))
    updater.start_polling()
    updater.idle()
if __name__ == '__main__':
    tmain()
    main()

```

					<i>ІАЛЦ.467100.007 Д4</i>	Арк.
Зм..	Арк.	№ докум.	Підпис	Дата		71

db.py

```
from resur import q

def get_question(ticket_number, question_number):

    question = q.questions[ticket_number][question_number]
    return question

def get_write_answer(ticket_number, question_number):

    write_answer = q.answers[ticket_number][question_number]
    return write_answer

def get_comment(ticket_number, question_number):

    comment = q.comments[ticket_number][question_number]
    return comment

def get_picture(ticket_number, question_number):

    picture = q.pictures[ticket_number][question_number]
    return picture

def get_number_of_choices(ticket_number, question_number):

    number_of_choices = q.choices[ticket_number][question_number]
    return number_of_choices
```

messages.py

```
# Повідомлення для користувача
PASSED = 'Іспит здано! Кількість правильних відповідей: '
FAILED = 'Іспит не складено! Можливо тобі варто потренуватися (/training)\n'
\
    'Кількість правильних відповідей: '
EXAM_STARTED = 'Іспит почався! Успіхів!'

# Команди
WELCOME = 'Ласкаво просимо в тренажер ПДР\n\n' \
    'Тут ти можеш тренувати свої знання в двох режимах:\n' \
    '1. Режим тренування (/training). Після кожного питання\n' \
    'ти дізнаєшся правильну відповідь і коментар до питання. \n' \
    '2. Режим іспиту (/exam). Тобі буде задано 20 питань.\n' \
    'В кінці ти дізнаєшся кількість набраних балів і результат іспиту\n\n'

# Кнопки бота
ONE = '1 '
TWO = '2 '
```

										Арк.
										72
Зм..	Арк.	№ докум.	Підпис	Дата	<i>ІАЛЦ.467100.007 Д4</i>					

```
THREE = '3'
FOUR = '4'
MENU = 'Повернутися в меню'
SOS = 'Коментар до питання'
NEXT = 'Наступне питання'
AGAIN = 'Перескласти екзамен'
```

models.py

```
from sqlalchemy import Column, Integer, Text, create_engine
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import sessionmaker

engine = create_engine('sqlite:///bot.db', echo =
True,connect_args={'check_same_thread': False})
Session = sessionmaker()
Base = declarative_base(bind=engine)

class User(Base):
    __tablename__ = 'users'
    id = Column(Integer, primary_key=True)
    name = Column(Text)
    user_name = Column(Text)

    def __repr__(self):
        return 'id=%r, name=%r, user_name=%r' % (self.id, self.name,
self.user_name)

class Ex(Base):
    __tablename__ = 'res_exam'
    id = Column(Integer, primary_key=True)
    k_mis = Column(Integer)
    ticket = Column(Integer)

    def __repr__(self):
        return 'id=%r, k_mis=%r, ticket=%r' % (self.id, self.k_mis,
self.ticket)

class Tr(Base):
    __tablename__ = 'tr'
    id = Column(Integer, primary_key=True)
    mis = Column(Integer)
    ticket = Column(Integer)

    def __repr__(self):
        return 'id=%r, mis=%r, ticket=%r' % (self.id, self.mis, self.ticket)
```

					<i>ІАЛЦ.467100.007 Д4</i>	Арк.
Зм..	Арк.	№ докум.	Підпис	Дата		73