

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається  
Завідувач кафедри  
\_\_\_\_\_ Скарга-Бандурова І.С.

« \_\_ » \_\_\_\_\_ 20\_\_ р.

ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА

ПОЯСНЮВАЛЬНА ЗАПИСКА

НА ТЕМУ:

Програмний модуль для автоматизації надання послуг в закладі  
громадського харчування

Освітній ступінь "бакалавр"  
Спеціальність 123 "Комп'ютерна інженерія"  
(шифр і назва спеціальності)

Керівник проекту:	_____	<u>М.В. Деркач</u>
	(підпис)	(ініціали, прізвище)
Консультант з охорони праці:	_____	<u>Я.О. Критська</u>
	(підпис)	(ініціали, прізвище)
Здобувач вищої освіти:	_____	<u>О.С. Гусейнова</u>
	(підпис)	(ініціали, прізвище)
Група:		<u>КІ-16Д</u>

Севєродонецьк 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки  
Кафедра Комп'ютерних наук та інженерії  
Освітній ступінь бакалавр  
Спеціальність 123 "Комп'ютерна інженерія"  
(шифр і назва)

ЗАТВЕРДЖУЮ:

Т.в.о. завідувача кафедри  
Сафонова С.О.  
«    »      20     р.

ЗАВДАННЯ  
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) БАКАЛАВРА

Гусейнової Олени Сергіївни

(прізвище, ім'я, по батькові)

1. Тема роботи: Програмний модуль для автоматизації надання послуг в закладі громадського харчування  
керівник проекту (роботи) к.т.н., доц. Деркач М.В.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від 30.04.2020 р. № 73/15.15

2. Термін подання студентом роботи: 11.06.2020

3. Вихідні дані до роботи: матеріали переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

Застосування інформаційних технологій у закладах громадського харчування.  
Вибір засобів розробки. Розробка програмного модулю автоматизації надання послуг в закладах громадського харчування. Охорона праці.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):  
електронні плакати

---

---

---

---

---

### 5. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	ст.викл. Критська Я.О.		

6. Дата видачі завдання 30.04.2020

Керівник

\_\_\_\_\_ (підпис)

Завдання прийняв до виконання

\_\_\_\_\_ (підпис)

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту(роботи)	Строк виконання етапів проекту ( роботи )	Примітка
1.	Отримання завдання	30.04.20	
2.	Огляд та аналіз літературних джерел	01.05.20 – 06.05.20	
3.	Аналіз актуальності теми розробки	06.05.20 – 10.05.20	
4.	Аналіз існуючих сервісів	11.05.20 – 15.05.20	
5.	Постановка завдання на розробку	16.05.20 – 17.05.20	
6.	Розробка програмного модулю	18.05.20 – 26.05.20	
7.	Оформлення пояснювальної записки ДП	27.05.20 – 08.06.20	
8.	Підготовка презентації	08.06.20 – 09.06.20	
9.	Підготовка доповіді	10.06.20 – 11.06.20	

Здобувач вищої освіти

\_\_\_\_\_ (підпис)

Керівник

\_\_\_\_\_ (підпис)

О.С. Гусейнова

(ініціали, прізвище)

М.В. Деркач

(ініціали, прізвище)

## РЕФЕРАТ

Пояснювальна записка містить: 83 сторінки, 24 рисунки, 6 таблиць, 2 додатки.

Мета роботи: розробка програмного модулю для автоматизації надання послуг в закладах громадського харчування.

Під час виконання диплому був проведений аналіз актуальності впровадження сучасних технологій для автоматизації надання послуг у сфері громадського харчування. У роботі розроблений програмний модуль, що автоматизує процес відправки замовлення на доставку їжі із закладу громадського харчування. Для адміністратора є можливість додавання нових товарів та відправлення розсилання користувачам про акції та нові пропозиції. Для розробки програмного модулю використовувалась мова програмування Python, бібліотека aiogram, Telegram Bot API, для розробки бази даних – бібліотека Gino, PostgreSQL.

Були розглянуті та описані норми охорони праці та безпеки в надзвичайних ситуаціях.

Ключові слова: ПРОГРАМНИЙ МОДУЛЬ, АВТОМАТИЗАЦІЯ, ПОСЛУГИ В ЗАКЛАДАХ ГРОМАДСЬКОГО ХАРЧУВАННЯ, МЕСЕНДЖЕР TELEGRAM, ЧАТ-БОТ, TELEGRAM BOT API.

Умови одержання дипломного проекту: СНУ ім. В. Даля, пр. Центральний 59-А, м. Сєвєродонецьк, 93400

## ЗМІСТ

ВСТУП.....	7
1 ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У ЗАКЛАДАХ ГРОМАДСЬКОГО ХАРЧУВАННЯ .....	22
1.1 Аналіз стану надання послуг у закладах громадського харчування .....	22
1.2 Автоматизовані послуги закладів харчування.....	22
1.3 Аналіз існуючих сервісів для надання послуг закладів харчування .....	22
1.4 Постановка завдання .....	22
Висновки до розділу 1 .....	22
2 ВИБІР ЗАСОБІВ РОЗРОБКИ .....	23
2.1 Вибір мови програмування .....	32
2.2 Бібліотека aiogram .....	32
2.3 Протокол HTTP .....	32
2.4 Telegram Bot API .....	32
2.5 Система управління базами даних .....	32
2.6 Середовище розробки PyCharm.....	32
Висновки до розділу 2 .....	32
3 РОЗРОБКА ПРОГРАМНОГО МОДУЛЮ АВТОМАТИЗАЦІЇ НАДАННЯ ПОСЛУГ В ЗАКЛАДАХ ГРОМАДСЬКОГО ХАРЧУВАННЯ .....	33
3.1 Загальна концепція розробки.....	33
3.2 Структура бази даних.....	39
3.3 Розробка програмного модулю .....	42
Висновки до розділу 3 .....	49
4 ОХОРОНА ПРАЦІ .....	50
4.1 Загальні питання з охорони праці .....	50
4.2 Аналіз стану умов праці.....	50
4.2.1 Вимоги до приміщення .....	51
4.2.2 Вимоги до організації місця праці .....	51

	6
4.2.3 Навантаження та напруженість процесу праці.....	53
4.3 Виробнича санітарія.....	54
4.3.1 Аналіз небезпечних шкідливих факторів при виробництві (експлуатації) виробу.....	54
4.3.2 Пожежна безпека.....	56
4.4 Гігієнічні вимоги до параметрів виробничого середовища.....	58
4.4.1 Освітлення.....	58
4.4.2 Шум та вібрація.....	59
Висновки до розділу 4.....	61
ВИСНОВКИ.....	62
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	64
Додаток А.....	68
Додаток Б.....	78

## ВСТУП

В епоху стрімкого розвитку комп'ютерних технологій завдання автоматизації процесу обслуговування людей у різних галузях є актуальним. Одна із перспективних та затребуваних - це галузь громадського харчування. Сучасний темп життя є дуже швидким та завантаженим. Задля економії часу, люди все частіше користуються послугами громадського харчування, особливо, послуга онлайн-замовлення.

У наш час стрімко збільшується кількість закладів громадського харчування. Зростання конкуренції стимулює власників до надання високого рівня обслуговування. Запорука успіху будь-якого закладу харчування - якість обслуговування. Повне задоволення різноманітних потреб клієнтів (та надання зручності) – ключ до успішного розвитку, зростання популярності та збільшення кількості відвідувачів. Це призводить до необхідності постійного пошуку нових варіантів удосконалення в сфері надання послуг. Зростають вимоги клієнтів щодо поліпшення якості послуг у закладах громадського харчування. У зв'язку з цим вкрай важливо підвищення зручності та швидкості обслуговування. Це можливо забезпечити шляхом використання сучасних програмних рішень - автоматизації надання послуг, що допоможе задовольнити потреби найвибагливіших клієнтів. Перевага автоматизації надання послуг у закладах громадського харчування очевидна, оскільки дає змогу замовляти послуги онлайн без черг, забезпечує користувача меню закладу в електронному вигляді, що дозволяє заощадити вільний час.

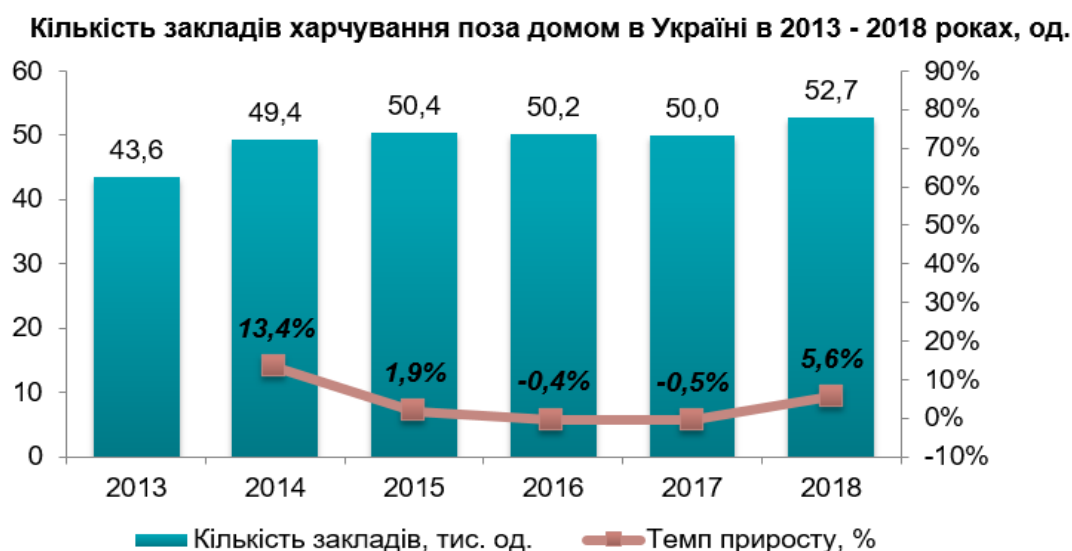
Створення програмного модулю для автоматизації надання послуг може зробити великий внесок у розвиток галузі обслуговування та вирішенні питання задовольнити потреби клієнтів, завдяки сучасних інформаційних технологій.

# 1 ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У ЗАКЛАДАХ ГРОМАДСЬКОГО ХАРЧУВАННЯ

## 1.1 Аналіз стану надання послуг у закладах громадського харчування

Сфера громадського харчування досить швидко розвивається. Кожного року на території України відкривається дедалі більше закладів, що насамперед викликано постійно зростаючою потребою людей у отриманні послуг харчування та цікавому проведенні відпочинку. Не зважаючи на нестабільний економічний стан економіки в Україні, заклади харчування користуються значним попитом у порівнянні з іншими місцями проведення дозвілля чи способами організації різних культурних заходів.

Проведений аналіз закладів харчування в Україні дає змогу побачити, що розвиток країни сприяє досягненню європейського рівня життя, а підприємці, що вклали гроші в дану галузь, мають збільшення власного прибутку. У порівнянні з відвідуванням фаст-фудів, ринок громадського харчування ще довго буде досягати повного насичення [1] (рис.1.1).



Джерело: дані Державної служби статистики, оцінка Pro-Consulting

Рисунок 1.1 – Статистичні дані кількості закладів харчування в Україні за 2013-2018 рр.



Сфера громадського харчування – є , насамперед, сферою надання різноманітних послуг клієнтам. Послуга, як результат економічної діяльності, направлена на задоволення фізіологічної потреби споживачів у спеціальних відведених для такого обслуговування місцях. Вона має відповідати санітарно-гігієнічним і законодавчим нормам. Діяльність закладів громадського харчування направлена на одержання прибутків в умовах функціонування, як окремих підприємств так і структурних підрозділів інших закладів ( торгівельних центрів (ТЦ), готелів і т.д.). Організація процесу надавання послуг харчування повинна забезпечувати не тільки задоволення фізіологічних потреб, а також організацію розваг та відпочинку, тобто задоволення соціокультурних потреб [2].

«Послуги, які надають заклади харчування, визначаються як:

- послуги замовлення харчування;
- послуги з виготовлення кулінарної продукції;
- послуги з реалізації продукції;
- послуги, направлені на організацію дозвілля;
- інформативні та консультативні послуги та ін.»

Зазвичай, розвиток закладів харчування різних типів залежить від економічної ситуації регіону. На формування цього сегменту ринку мають вплив також соціальні, демографічні та культурні чинники. Економічна ситуація в нашій країні впливає так, що активно розвиваються заклади із середнім цінами на послуги і таким чином лідерами даного сегменту є кафе, кофейні, чайні, заклади фаст-фуду. Чинники другого розряду формують сегмент ринку підприємств, де більш чітко виражене надання соціально-культурних послуг, які є складовими послуг харчування.

Так, наприклад, підрозділи громадського харчування на автомобільній заправній станції (АЗС) займають 20,3% ринку, а ресторани, що розміщені в готелях - усього 3,9%.

**Структура ринку харчування поза домом в Україні за сегментами у 2018 році  
(з урахуванням оцінки частки ФОП),%**



*Джерело: оцінка Pro-Consulting*

Рисунок 1.2 – Статистичні дані структури ринку харчування в Україні за 2018 р.

Основними складовими ефективної роботи закладу громадського харчування є якість обслуговування, перелік послуг, кухня, цінова категорія та атмосфера. Тісна взаємодія даних складових є запорукою успіху. Однак, головна задача ефективної діяльності визначається як задовольнити потребам найвибагливіших споживачів. Якщо клієнти не отримують задоволення то все інше немає значення. Виникає необхідність у постійному зростанню якості надання послуг клієнтам закладами громадського харчування задля конкурентоспроможності підприємств. Це вимагає урахування особливостей способу життя потенційних клієнтів та розширення сервісу відповідно до їх потреб. У наш час із високим темпом життя все більше людей мають завантажений графік. Багато працівників бізнес-середовища заради економії часу вважають за краще починати свій день у ресторанах, кафе або замовити їжу додому.

Зростає рівень конкуренції. Підприємцям, аби утриматися в цій сфері, необхідно постійно вдосконалювати роботу, запроваджувати найсучасніші технології, методики та методи не лише на рівні управління закладом, але й на

рівні обслуговування клієнтів [3]. В ресторанному господарстві введення інновацій можна визначити як виробничі (саме виробництва продуктів харчування), а також — невиробничі (надання послуг із організування харчування та додаткових послуг). Впровадження інновацій у сфері громадського харчування дають переваги як виробникам, так і споживачам [4].

Упровадження інновацій впливає на підвищення рівня задоволення потреб найвибагливіших клієнтів, що збільшує конкурентоспроможність закладу, а як результат – підвищує прибутковність закладу. Це обумовлено тим, що інновації є результатом наукової діяльності, що втілюються в реальні речі і процеси та дають змогу людству виходити на новий більш якісний рівень розвитку та існування.

Внаслідок тенденцій у запровадженні інноваційних методів в обслуговуванні клієнтів, визначаються такі можливості: розширення асортименту певних видів продукції, приваблення клієнтів розмаїтістю додаткових послуг, уведення спеціальних пропозицій та уведення системи знижок для заохочення клієнтів. Насамперед, це має соціальний ефект, так як часто важливу роль грають не ціни, а сама зацікавленість людей та оригінальність підходу.

Використання сучасних інформаційних технологій передбачає використання інформаційно-комп'ютерних технологій (ІКТ) для автоматизації надання послуг у закладах громадського харчування, що дає змогу значно спростити, розширити і поліпшити сервіс для вибагливих клієнтів. Це важливий крок у розвитку галузі в цілому, зокрема підвищення рівня обслуговування клієнтів та спосіб вирішення сучасних проблем в області надання послуг клієнтам [5].

Одна із основних ІТ-інновацій для ресторанного бізнесу — це запровадження комплексних Інтернет- і телекомунікаційних технологій для взаємодії з клієнтами.

Інформаційні технології дозволяють використовувати нові можливості, які допомагають оптимальним шляхом із мінімальними затратами, донести

необхідну інформацію зацікавленим споживачам. Сьогодні Інтернет можна розглядати не лише як універсальне середовище передачі інформації, але і як спосіб розповсюдження реклами задля заохочення клієнтів. Популярність Інтернету виростає з кожним днем, і вже сьогодні Інтернет-простір, за порівнянням з традиційними СМІ, є куди більш вигідним місцем для розміщення реклами [6].

Сьогодні більш ніж 60% українців є активними користувачами інтернету. Це представники різних вікових категорій. Причому мобільний телефон виявився найпоширенішим пристроєм для взаємодії з контентом (70%). Інтернет для сучасних людей — уже не тільки джерело розваги, він є частиною їх повсякденного життя, зручним і швидким засобом для вирішення більшої частини буденних питань [7].

Сьогодні Інтернет є одним із найбільш ефективних способів заохочення клієнтів [8]. Як наслідок, дедалі частіше пропонуються надання послуг онлайн. Популярні такі послуги - доставки замовлень, здійснених по телефону чи мережі Інтернет, перегляд онлайн-меню та бронювання столиків онлайн.

## **1.2 Автоматизовані послуги закладів харчування**

*Використання електронного меню.* За статистичними даними 93% гостей починають свій похід в ресторан з пошуку меню в Інтернеті [9]. Кожен може ознайомитися з меню ресторану або кафе в спокійній обстановці вдома чи в офісі з комп'ютера та мобільного телефона. У користувачів є можливість переглянути зміст меню та всі доступні страви і напої. Зазвичай онлайн-меню представлені у вигляді яскравих привабливих фото з описом інгредієнтів. Великою перевагою є те, що клієнти можуть не тільки ознайомитися із меню закладу, а й замовити страви до дому чи в офіс. Доступний вибір параметрів, що впливають на ціну. Це є дуже зручно для людей сьогодні, адже такий спосіб харчування економить час. Також можна зробити онлайн замовлення і, з приходом в заклад, страви вже будуть на столі [10].

Переваги онлайн-меню для закладу також дуже значні. Онлайн-меню дозволяє закладам розширювати базу клієнтів і, як наслідок отримувати більший прибуток.

*Послуга доставки.* Попит на послуги доставки їжі швидко росте. Хоча, заклади громадського харчування і залишаються улюбленим способом проводити час парами, сім'єю і друзями протягом десятиліть, але тенденція останніх років змінилася: люди дуже люблять замовляти їжу, не виходячи з власного будинку.

Сьогодні доставка їжі додому є досить затребуваною послугою, яка дозволяє сучасній людині економити дорогоцінний час.

За результатами проведеного маркетингового дослідження був зроблений висновок про подальше продовження розвитку ринку доставки харчової продукції на тлі бурхливого зростання інтернет-торгівлі продуктами харчування. Немалу важливу роль у розвитку ринку доставки харчової продукції відіграло збільшення частки інтернет-користувачів, розпочинаючи із 2010 року кількість яких зросла з 30% аж до 63% відповідно до підрахунків 3 кварталу 2018 року. А розпочинаючи з 2016 року майже в 2 рази збільшилась кількість запитів на доставку їжі в Інтернеті.

На сьогодні на даному ринку виділяють шість головних сегментів. Окрім надання послуг звичайної доставки із ресторанів та кафе, на ринку з'являються послуги із доставки додому наборів їжі на день чи тиждень, наборів продуктів для власного приготування окремих страв. Не зважаючи на присутність значної кількості суміжних сегментів, все ж таки основними підприємствами на цьому ринку так і залишаються ресторани і кафе з наданням послуги доставки та доставки харчової продукції із супермаркетів [11].

Переваги надання послуг доставки їжі:

- збільшення продаж;
- розширення кількості клієнтів;
- вихід на новий ринок;

– відсутність необхідності розширення ресторану і збільшення кількості місць.

Оскільки люди завжди набувають все нові і нові звички, прагнучі до задоволення своїх потреб найзручнішим для них способом, можна зробити висновок, що звичка замовляти їжу онлайн дуже скоро буде притаманна сучасним людям.

*Бронювання столиків онлайн.* Як уже вказувалося вище, сучасний темп та завантаженість графіку стимулює людей використовувати нові технології для економії дорогоцінного часу. Однією із таких можливостей – бронювання столиків в закладах громадського харчування онлайн. Бронювання столиків онлайн дуже економить час відвідувачеві, що дає йому можливість заочно детально познайомитися з закладом, кухнею і ціновою політикою.

В США значна частина клієнтів кафе та ресторанів користуються можливістю бронювати столик заздалегідь через онлайн-сервіси. Уже сьогодні дана послуга набирає популярність також на території України. Це обумовлене тим, що активні молоді люди, що займаються кар'єрою, все більше цінують свій час, через що прагнуть зекономити його у всьому. Бронювання столиків в кафе чи ресторані значно скорочує їм час.

Розміщення інформації на сервісах онлайн-бронювання дає змогу кафе чи ресторану розширити базу постійних клієнтів. Запровадження автоматизації процесів надання послуг дозволяє зменшити для менеджерів навантаження, пов'язаного із черг на бронювання столиків та зайняти кориснішою роботою. Відтак, значно зростає якість обслуговування, збільшується час на реалізацію нових ідей і т.п.

Сьогодні системи онлайн бронювання столиків в кафе і ресторанах все ще розвивається. Проте, вони вже зараз привертають увагу все більшої кількості людей. Ресторани та кафе самі йдуть на контакт з подібними агрегаторами, оскільки з їх сторінок до них приходять більше клієнтів, ніж з офлайна.

### 1.3 Аналіз існуючих сервісів для надання послуг закладів харчування

Постійно зростаюча популярність надання послуг онлайн стимулює власників до використання сучасних Інтернет-сервісів. Аналіз використання ІТ в залученні клієнтів закладів громадського харчування показав, що найчастіше користуються такими технологіями [12, 13]:

- сайт-візитка;
- мобільний додаток;
- просування в соціальних мережах, контекстна реклама і SMM;
- чат-бот.

*Сайт-візитка* - затребуваний майданчик для розміщення візуального і текстового контенту, а також для оптимізації роботи сайту та залучення трафіку. Блог дозволяє розкривати відвідувачам будь-які події закладу, тим самим підвищуючи їх лояльність. Наявність власного сайту в Інтернеті, де можна не тільки знайти інформацію про заклад, а й замовити страви чи зарезервувати столик – є необхідністю для сучасної роботи кафе. Створення мобільної версії сайту і розробці мобільного додатку буде корисно для тих, хто часто користується інтернетом завдяки телефону. В даному випадку великий вплив має психологічний фактор - якщо на мобільному пристрої потенційного споживача сайт відображається так, що йому незручно знаходити всю потрібну інформацію, настає момент відмови від пропозиції ресторану.

Як приклад можна розглянути сайт українського ресторану у м. Києві «Первак» (рис. 1.3) [14].

Наявний наступний функціонал:

- можливість зарезервувати столик у обраному залі;
- перегляд меню закладу онлайн;
- можливість оформити замовлення доставки;
- перегляд інформації про заклад.

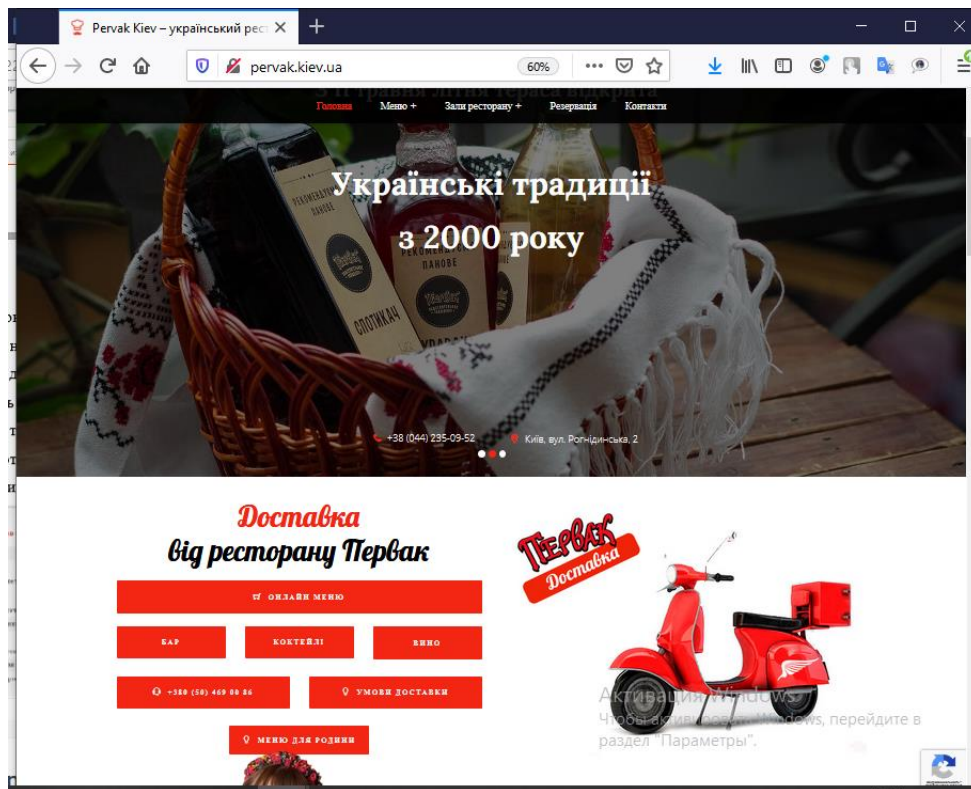


Рисунок 1.3 – Головна сторінка сайту

#### Переваги сайту:

- адаптація для мобільних пристроїв;
- цікавий та красивий дизайн;
- відсутність реклами;
- простий та зрозумілий інтерфейс.

#### Недоліки:

- необхідне постійне підключення до Інтернету;
- пошук необхідної інформації здійснюється за допомогою значної кількості кліків;
- відсутність авторизації користувача, кожного разу для замовлення необхідно вводити свої дані;
- мінімальна сума замовлення – 300 грн, що може бути не зовсім зручним для клієнтів;
- відсутність заохочення до покупки.



У наш час особливо популярні мобільні технології. Їх використання є надзвичайно важливим для впровадження ІТ-інновацій, адже одним з основних завдань інформаційних технологій – спростити та удосконалити життя людей. Майже кожен з нас має власний смартфон з доступом до. Створення *мобільного додатку* для закладів харчування є вигідною, актуальною та корисною задачею у наш час.

Для прикладу можна описати мобільний додаток мережі кав'ярень «Lviv Croissants» (рис. 1.4).

Додаток має такі можливості:

- перегляд новин на головній сторінці;
- перегляд меню;
- перегляд розташування закладів на карті;
- можливість додавати страви і напої в «Обране» для швидкого доступу;
- можливість отримання бонусів за користування додатком;
- форма авторизації користувача.

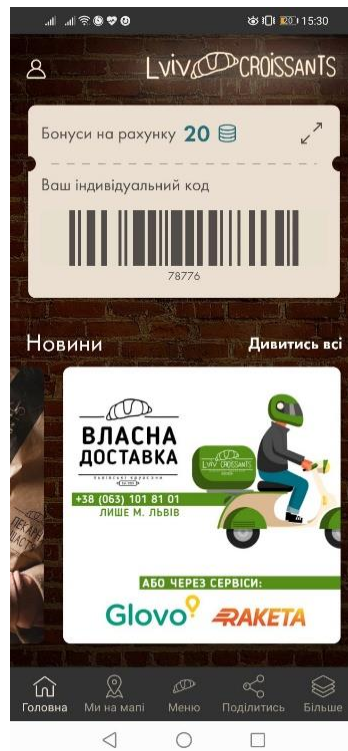


Рисунок 1.4 – Головне вікно додатку

### Переваги:

- зручно користуватися мобільним;
- красивий та простий дизайн;
- відсутність реклами;
- можна користуватися без доступу до Інтернету;
- можливість зворотного зв'язку у вигляді форми для коментарів.

### Недоліки:

- пошук необхідної інформації здійснюється за допомогою значної кількості кліків;
- немає можливості оформлення замовлення доставки, хоча така функція є у кав'ярні.

Однак сайти та додатки вимагають певних фінансових затрат. Тому одним із вигідних інструментів для просування послуг – є соціальні мережі.

Особливу увагу варто приділити технологіям SMM - Social Media Marketing - просування товарів і послуг в соціальних мережах, які сприймаються маркетингом, як соціальні медіа. За даними на 2018 рік найпопулярнішою мережею світу за даними [15] є Facebook, на другому місці знаходиться Instagram [16,17] (рис. 1.5).

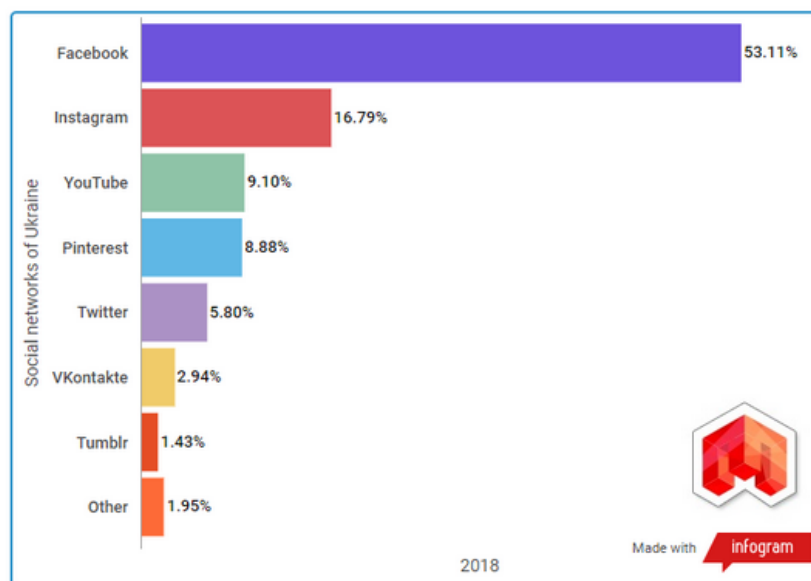


Рисунок 1.5 – Статистика соціальних мереж в Україні

Безсумнівно, в даний час соціальні мережі є одним з найефективніших способів просування товарів і послуг.

*Соціальні мережі* - ідеальний майданчик для комунікації з клієнтами, тому, саме працюючи над «близьким» споживачеві контентом, представники громадського харчування можуть підвищити відвідуваність своїх закладів і поліпшити імідж. Комунікація із споживачами в соціальних мережах може бути тематично налаштована під концепцію закладу.

Instagram - потужний маркетинговий інструмент. Майже у всіх активних користувачів смартфонів встановлено цю програму, багато хто використовує його для перегляду постів у стрічці. Ведення сторінки в Instagram надає бізнесу такі можливості: різноманіття контенту, взаємодія з блогерами і вплив на їх аудиторію, контекстну рекламу (рекламу, зміст якої залежить від інтересів користувача), проведення трансляцій в режимі онлайн і багато іншого.

Розглянемо сторінку @terrassa\_spb у соціальній мережі Instagram (рис. 1.6).

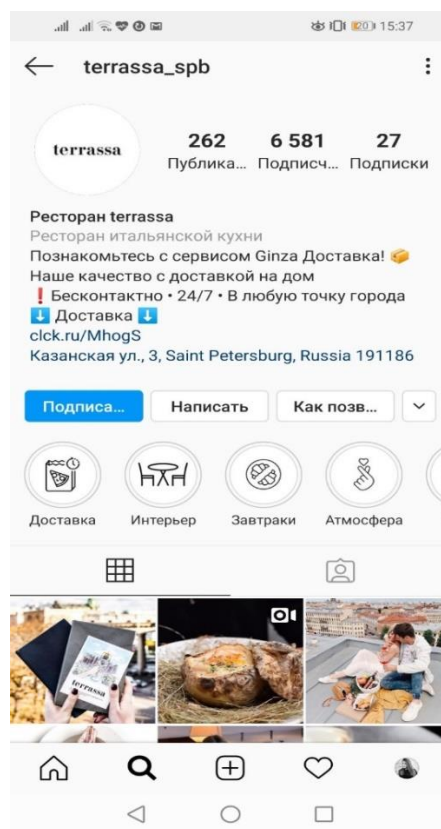


Рисунок 1.6 – Сторінка @terrassa\_spb

Підписники мають змогу переглядати красиві фото страв із меню, ознайомитися із подіями закладу та новинами. Деякі пости демонструють меню сайту у вигляді фото.

У порівнянні з іншими методами, такий має переваги:

- можливість підписуватися на потенційних клієнтів, тим самим розширювати цільову аудиторію;
- ознайомлення підписників із красивими фото страв та подіями закладу.

Такий спосіб залучення нових клієнтів є досить успішним, але для підвищення прибутку цього замало. У потенційних клієнтів є можливість перегляду красивих фото страв, але соц-мережа не має додаткових функцій для процесу замовлення. Є лише вказаний телефон кафе в описанні до сторінки (адреса сайту кафе). Але не всім користувачам до вподоби набирати номер на мобільному та робити замовлення, не маючи перед очами меню. Тому для найкращого результату у підвищенні прибутку, необхідне впровадження додаткового сервісу.

Окремої уваги заслуговує використання в бізнесі месенджерів - з англійського «кур'єр» чи «зв'язковий». Це спеціальні програми, що забезпечують миттєвий обмін повідомленнями між користувачами. Їх головна перевага полягає саме в швидкості обміну повідомленнями. Сьогодні деякі месенджери (наприклад, Telegram, Viber або Facebook Messenger) мають функції створення окремих каналів для оперативного донесення актуальних новин до цільової аудиторії. Також деякі сервіси дозволяють створювати чат-ботів, які здатні частково виконувати функціонал call-центрів і служб підтримки. Бот - похідне від слова «робот» - це програма, яка виконує будь-які дії через такий же інтерфейс, що і звичайний користувач.

Використання *чат-боту* у наданні послуг – є дуже актуальним та перспективним способом залучення клієнтів у закладах громадського харчування. Преваги чат-боту перед іншими сервісами у тому, що для користування він не потребує скачування чи реєстрації, адже сьогодні

присутність месенджера в телефоні – є необхідністю. Знайомий та зручний інтерфейс, простота використання – одні із головних переваг.

Прагнучі стати більш персоналізованим для клієнтів, бренди переносять розмову в приватні повідомлення. Такий обмін повідомленнями сприяє оптимізації обслуговування клієнтів та надання допомоги в продажах. Оформлення замовлення в приватних повідомленнях стає однією з найактуальніших тенденцій в 2020 році в області цифрового маркетингу [18-20]. А весь процес користування чат-ботом проходить в діалоговому вікні месенджера, що є дуже зручним для користувачів. Також складається враження, що користувач спілкується з людиною.

Функціонал месенджера дуже широкий. Доступні такі важливі функції надання послуг як:

- перегляд онлайн-меню закладу;
- замовлення доставки;
- можливість бронювання столиків;
- перегляд новин та подій закладу;
- можливість відправляти відгуки та побажання.

Також дуже зручним є те, що чат-бот допомагає автоматизувати ці процеси. І ще однією значною перевагою є те, що чат-бот може запам'ятовувати які саме замовлення обирав користувач і, виходячи з цього, пропонувати подібний набір страв. Це сприяє збільшенню вірогідності, що користувач захоче зробити наступне замовлення. Також можлива функція відгуків та пропозицій, де користувачі можуть залишати свої враження та пропозиції щодо поліпшення сервісу. Це може значно вплинути на репутацію та кількість клієнтів у закладах громадського харчування.

Із проведеного аналізу використання інформаційних технологій для надання послуг клієнтам закладами громадського харчування можна зробити висновок, що задля найкращого результату потрібно використовувати сервіс, який, насамперед, є зручним, інформативним та безумовно функціональним.

## **1.4 Постановка завдання**

Підставою до розробки програмного модулю автоматизації надання послуг в закладах громадського харчування є підвищення попиту клієнтів на користування онлайн-сервісів для задоволення власних потреб.

Об'єктом дослідження – є автоматизований процес надання послуг клієнтам в закладах громадського харчування.

Предмет дослідження – методи й засоби автоматизації надання послуг клієнтам в закладах громадського харчування.

Мета роботи – розробка програмного модулю для розширення функціональних можливостей надання послуг клієнтам в закладах громадського харчування, шляхом їх автоматизації.

Основними задачами є аналіз предметної області з метою дослідження актуальності, огляд наявних проблем та оцінка їх можливих рішень, визначення засобів розробки та власне реалізація програмного модулю.

Програмний модуль дозволить підвищити ефективність процесу обслуговування, завдяки впровадженню цифрового меню, забезпечення можливості онлайн онлайн замовлення доставки. Це призведе до підвищення продуктивності роботи закладу харчування, скорочення черг та часу очікування, зростання потоку клієнтів, за рахунок впровадження чат-боту до веб-додатку.

## **Висновки до розділу 1**

Проведено аналіз сучасного стану застосування інформаційних технологій в закладах харчування та існуючих сервісів для надання послуг у таких закладах, який показав, що розробка програмного модулю з підтримкою чат-боту може підвищити ефективність автоматизації наданих послуг.

## 2 ВИБІР ЗАСОБІВ РОЗРОБКИ

### 2.1 Вибір мови програмування

Python – є однією із найпопулярніших мов програмування сьогодні. Вона є високорівневою мовою загального призначення, орієнтованою на підвищення продуктивності розробника та читабельність коду. Дана мова програмування використовується для вирішення різних завдань і надає такі можливості: підтримку ООП, динамічність і крос-платформеність. Розробка Python була розпочата Гвідо Ван Россум в середині 1990-х років.

Архітектурні особливості мови Python – динамічна типізація, підтримка багатопотокових обчислень, автоматичне управління даних. Значним плюсом є підтримка розбиття програми на модулі, які можуть об'єднуватися в пакети. У будь-якої мови як для спілкування, так і для програмування, є обов'язкові складові - словник і синтаксис. Мова програмування Python організована точно так, воє має синтаксис для формування виразів та словник - набір функціональності, представлений у вигляді стандартної бібліотеки і модулів.

Особливість Python полягає у тому, що не зважаючи на простоту коду, рівень абстракції даної мови значно вищий, наприклад, ніж у таких мов як C++ та Java, що дозволяє передавати таку кількість інформації в меншому обсязі коду. Завдяки своїй лаконічності та простоті дана мова програмування дозволяє програмісту швидше опанувати синтаксис. А за рахунок простоти написання коду подальший супровід написаних на ній програм є легшим і зручнішим у порівнянні з такими мовами як C++ та Java. Перевага для бізнесу полягає у тому, що скорочуються витрати і збільшуються продуктивність праці співробітників [22].

У таблиці 2.1 представлена порівняльна характеристика мови програмування Python з C++ та Java.

Таблця 2.1 - Порівняльна характеристика мов програмування

	Python	Java	C ++
Швидкість виконання програми	середня	середня	висока
Використання пам'яті	споживає невелику кількість пам'яті	споживає велику кількість пам'яті	Споживає невелику кількість пам'яті
Простота написання коду	проста	складна	складна
Об'єм коду	малий	великий	великий
Поріг входження (легкість вивчення новачками)	легко	середня	важко
Масштабованість	присутня	присутня	присутня
Кросплатформеність	висока	висока	середня
Типізація	динамічна	статична	статична

Передбачається, що програми, написані на Python, працюють декілька повільніше, ніж програми на Java, однак розробка їх займає набагато менше часу. Через динамічну типізацію виконання програми на Python йде важче, ніж у випадку з Java. Наприклад, обчислюючи вираз  $a + b$ , програма повинна спочатку визначити тип об'єктів  $a$  і  $b$ , адже він невідомий при компіляцію. Java, в свою чергу, може здійснювати ефективне додавання цілих чисел або чисел з плаваючою комою, але вимагає декларування змінних  $a$  і  $b$  і не дозволяє перезавантаження оператора  $+$ , наприклад, певними користувачем класами. Виходячи з цього, Python краще підходить для використання як інтегруюча мова, в той час як Java швидше характеризується як мова низькорівневого виконання.

Як і Python, C ++ - це мова програмування загального призначення, що використовується в різноманітних завданнях. Мова програмування є об'єктно-орієнтованою, отже, здатна працювати швидше і ефективніше.

Мова C ++ є протилежністю Python з точки зору простоти. Насправді вона вважається однією з найскладніших мов програмування. Її надскладний синтаксис дозволяє C ++ бути надзвичайно гнучкою - це основна причина, чому її часто вибирають для роботи з проектами, які розробляються для декількох різних платформ одночасно.



### Переваги Python:

- висока швидкість виконання програм;
- в стандартних бібліотеках Python можна знайти рішення для роботи з протоколами Інтернету, HTTP , електронною поштою, FTP, базами даних;
- скрипти, написані мовою Python можуть виконуватися на більшості сучасних операційних системах (ОС). Це дає можливість Python застосовуватися в різних областях;
- над розробкою Python працювали тисячі ентузіастів з усього світу та підтримують такі компанії як Google, Mozilla, Facebook, Yandex, RedHat, Telegram.

Мова програмування Python є універсальною – вона може використовуватися в самих різних областях розробки програмного забезпечення (клієнт-сервер, Web-додатки, ігри, комплексні системи розрахунків, скрипти автоматизації, чат-боти та багато іншого) та в будь-якій предметній області.

## 2.2 Бібліотека aiogram

Aiogram – це бібліотека для мови програмування Python для Telegram Bot API, написана на asyncio и aiohttp. Asyncio – це бібліотека для написання паралельного коду. Ця бібліотека асинхронна, тому більш продуктивніша та швидша. Має детальну документацію та багато прикладів. Для того, щоб встановити бібліотеку, необхідно виконати просту команду:

```
python -m pip install -U aiogram
```

Для використання бібліотеки необхідно імпортувати модулі:

```
import logging
from aiogram import Bot, Dispatcher, executor, types
```

Усі функції в даній бібліотеці є асинхронними, тому перед ними необхідно писати кодове слово `async`:

```
@dp.message_handler(commands=['start', 'help'])
async def send_welcome(message: types.Message):
    await message.reply("Hi!\nI'm EchoBot!\nPowered by aiogram.")
    await message.reply("Привет!\nЯ бот!.")
```

Await – кодове слово, яке означає що функція може запускатися одночасно.

Перевагою асинхронного програмування є можливість відправлення одночасно декілька повідомлень, не дочекавшись поки відправиться перше. Таким чином збільшується швидкодія роботи.

## 2.3 Протокол НТТР

НТТР - широко розповсюджений протокол передачі даних. Його використовують для передачі гіпертекстових документів (документів, які містять посилання для організації переходу до інших документів). Протокол НТТР лежить в основі обміну даними в Інтернеті. НТТР є протоколом клієнт-серверної взаємодії, що означає ініціювання запитів до сервера самим одержувачем, зазвичай веб-браузером (рис.2.1).

Абревіатура НТТР розшифровується як HyperText Transfer Protocol, «протокол передачі гіпертексту». За специфікацією OSI, НТТР є протоколом прикладного (верхнього, 7-го) рівня.

Протокол НТТР припускає використання клієнт-серверної структури передачі даних. Клієнтська програма формує запит та відправляє на сервер, після чого серверне ПЗ обробляє цей запит, формує відповідь та відправляє його назад клієнту. Далі клієнтська програма може продовжити відправлення інших запитів, що будуть оброблятися аналогічним способом.

Завдання, яке вирішується за допомогою протоколу НТТР - обмін даними між призначеним для користувача додатком, що здійснює доступ до веб-ресурсів (зазвичай це веб-браузер) і веб-сервером. На даний момент саме завдяки протоколу НТТР забезпечується робота Всесвітньої павутини.

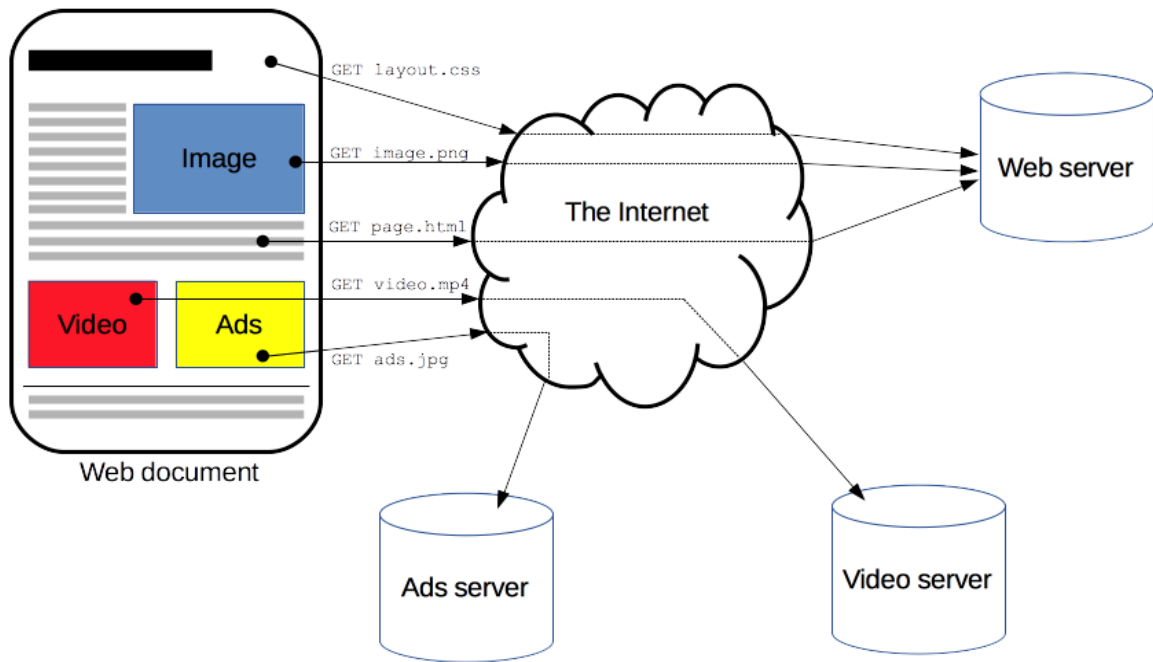


Рисунок 2.1 – Протокол клієнт-серверної взаємодії HTTP

API багатьох програмних продуктів також має на увазі використання HTTP для передачі даних - самі дані при цьому можуть мати будь-який формат, наприклад, XML або JSON.

Взаємодія з чат-ботами засновано на простих HTTP-запитах.

## 2.4 Telegram Bot API

Bot API представляє HTTP-інтерфейс, призначений для роботи з чат-ботами в месенджері Telegram. Чат-бот - це спеціальний аккаунт, створений для взаємодії з користувачем, використовуючи автоматичне оброблення та відправлення повідомлення.

Оновлення від бота можна отримати двома способами, які відрізняються за логікою роботи:

- long pulling – створений додаток автоматично опитує сервер Telegram на наявність будь-яких оновлень для бота;

– webhook - сервер Telegram сам сповіщає додаток на сервері про наявність нових оновлень. Вхідні оновлення зберігаються на сервері, поки їх не оброблять, однак не довше 24 годин.

Не залежно від методу отримання оновлення, у відповідь відправляється об'єкт Update, в JSON форматі. Всі запити до Telegram Bot API здійснюються через HTTPS в такому вигляді:

*[https://api.telegram.org/bot<token>/НАЗВА\\_МЕТОДУ](https://api.telegram.org/bot<token>/НАЗВА_МЕТОДУ)*

Принцип роботи взаємодії чат-бота й користувача зображений на рисунку 2.2.

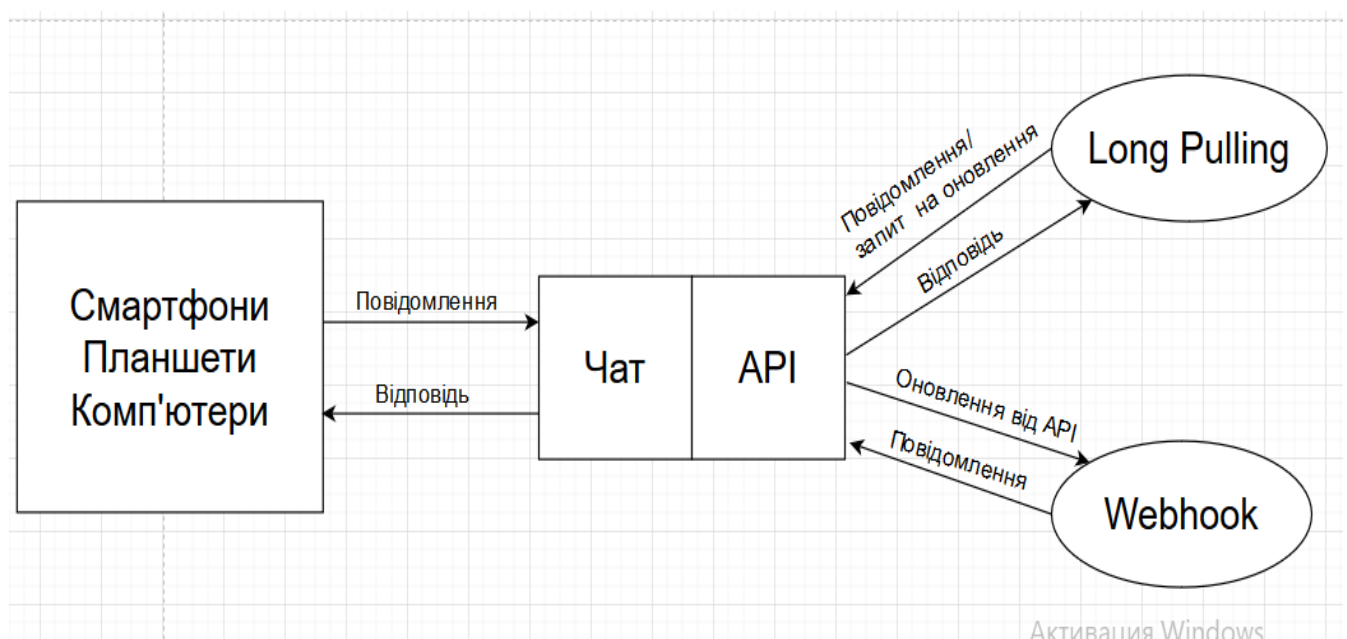


Рисунок 2.2 – Принцип роботи чат-боту на платформі Telegram

Приклади доступних для API методів описані нижче:

- `getUpdates` - цей метод використовується для отримання оновлень по технології Long polling. Під оновленнями мається на увазі якась дія, вчинена з ботом – як приклад - отримання повідомлення від користувача. При тривалому опитуванні клієнт запитує інформацію від сервера точно так само, як і при звичайному опитуванні, але сервер може не відповісти негайно. Якщо сервер не

має нової інформації для клієнта при отриманні опитування, замість надсилання порожньої відповіді, сервер тримає запит відкритим і чекає, коли інформація про відповідь стане доступною. Після отримання нової інформації сервер негайно надсилає клієнту відповідь HTTP/S Requests, виконуючи відкритий запит HTTP/S Requests. Після отримання відповіді сервера клієнт часто негайно видає інший запит сервера. Таким чином усувається звичайна затримка відповіді (час між тим, коли інформація вперше стає доступною при наступному запиті клієнта), інакше пов'язане з опитуванням клієнтів.

- setWebhook - метод приєднує до бота url домену, де знаходиться запусканий бот. Цей метод необхідний для завдання URL вебхука, на який бот зможе відправляти нові оновлення. При отриманні оновлень кожного разу на цю адресу буде відправлятися запит HTTPS POST з переформованим в JSON об'єктом Update. Якщо запит невдалий до сервера клієнта спроба буде повторюватись декілька разів;

- sendMessage - метод відправляє текстове повідомлення в клієнт Telegram;
- sendLocation - метод відправляє повідомлення з координатами локації користувачів клієнт Telegram;

- getFile - метод повертає файл по його імені та ін. Допускаються POST і GET запити.

Щоб передати параметри в Bot API є 4 способи: запит в URL, application / x-www-form-urlencoded, application / json (не прийнятний для завантаження файлів), multipart / form-data (для завантаження файлів).

## **2.5 Система управління базами даних**

Для роботи з базами даних (БД) використовувалась популярна вільна об'єктно-реляційна система управління базами даних (СУБД) PostgreSQL. PostgreSQL - одна з найбільш просунутих і широко використовуваних СУБД. PostgreSQL базується на мові SQL.

СУБД PostgreSQL відрізняється високою надійністю та хорошою продуктивністю роботи. Вона дозволяє працювати не тільки зі структурованими і нормалізованими, а й зі слабоструктурованими даними (в форматах json / jsonb), при цьому ці дані індексуються і робота з ними дійсно зручна. Особливу цінність ці можливості знаходять при розробці систем, де моделі даних повинні бути кастомізованими в режимі реального часу на рівні GUI - в інтернет-магазинах, де у товарів може бути безліч різноманітних характеристик, в системах електронного документообігу та в інших бізнес-додатках. При цьому в розробці складних сайтів і веб-додатків PostgreSQL випереджає по використанню MySQL і MariaDB. Більшість фреймворків (наприклад, Ruby on Rails, Yii, Symfony, Django) підтримують використання PostgreSQL в розробці.

PostgreSQL має наступні переваги:

- високопродуктивні і надійні механізми транзакцій і реплікації;
- підтримка БД необмеженого розміру;
- розширювана система вбудованих мов програмування і підтримка завантаження C-сумісних модулів;
- наслідування;
- розширюваність (можливість створювати нові типи даних, типи індексів, мови програмування, модулі розширення, підключати будь-які зовнішні джерела даних).

Для зручності роботи з БД PostgreSQL для мови Python є бібліотека gino. Дана бібліотека допомагає писати і виконувати SQL в асинхронній програмі. Замість безпосередньої взаємодії СУБД з неопрацьованим SQL, можна отримати доступ до своїх даних через дружельюбний API. Бібліотека має досить гнучкий інтерфейс.

Якщо взяти будь-які дані однакової структури і обсягу, то сортування, угруповання і агрегація цих даних в Python буде на порядок повільніше, ніж в СУБД. Навіть, якщо в Python ці дані будуть розташовуватися в пам'яті. Але при передачі результату обробки даних від СУБД в Python код буде на два порядки довше, ніж обробка їх в Python. Саме введення / виведення – це сама повільна

частина. Настільки повільна, що на її тлі все інше – несуттєві витрати. Цю проблему і вирішує асинхронність, дозволяючи програмі замість простоювання виконувати будь-які корисні дії в момент очікування введення / виведення. Але це працює лише у тому випадку, коли абсолютно весь код асинхронний. Якщо один виклик блокується, асинхронність решти ланцюжка викликів марна.

## 2.6 Середовище розробки PyCharm

PyCharm – це інтегроване середовище розробки, призначене для мови програмування Python. Надає зручні засоби для аналізу коду, графічний відладчик, інструмент для запуску юніт-тестів і підтримує веб-розробку на Django. PyCharm розроблений компанією JetBrains на основі IntelliJ IDEA. Може працювати на операційних системах Windows, Mac OS X і Linux.

PyCharm виконує перевірку коду на льоту, автоматичне доповнення, навігацію по кодові, забезпечує безліч рефакторінгів, тобто можливості перейменувати. Також дане IDE представляє першокласну підтримку для Python, JavaScript, TypeScript, CSS, популярних мов і багато чого іншого.

IDE PyCharm випускається в двох варіантах - безкоштовна версія PyCharm Community Edition і підтримує більший набір можливостей PyCharm Professional Edition.

Основні можливості:

- підсвічування синтаксису і помилок;
- відображення файлової структури проекту, швидкий перехід між файлами, класами, методами і використанням методів;
- рефакторінг: можливість перейменувати, введення нової змінної, введення констант і т.д.;
- є вбудований швидкодіючий відладчик для Python;
- інструменти для веб-розробки з використанням фреймворку Django;
- вбудовані інструменти для unit-тестування;

- підтримка систем контролю версій: загальний користувальницький інтерфейс для Git, Subversion, Perforce і CVS із підтримкою списків змін і злиття.

## **Висновки до розділу 2**

У другому розділі були розглянуті методи та інструменти для розробки програмного модулю. У якості мови програмування була обрана Python та бібліотека aiogram. Для зручної роботи з базами даних обрана СУБД PostgreSQL.



### 3 РОЗРОБКА ПРОГРАМНОГО МОДУЛЮ АВТОМАТИЗАЦІЇ НАДАННЯ ПОСЛУГ В ЗАКЛАДАХ ГРОМАДСЬКОГО ХАРЧУВАННЯ

Для досягнення поставленої мети, перш за все, необхідно:

- розробити концепцію функціонування програмного модулю;
- розробити БД;
- розробити програмний модуль.

#### 3.1 Загальна концепція розробки

Головна мета розробки програмного модулю – за рахунок чат-боту автоматизувати процес надання послуг закладу громадського харчування, а саме кафе «Жиравф» міста Рубіжне. Таке рішення безпосередньо сприятиме розширенню клієнтської бази кафе.

Модуль є програмним продуктом, який для збільшення переваг у сфері наданні послуг клієнтам закладом громадського харчування окрім web-додатку підтримує чат-бот. На рисунку 3.1 зображено структура програмного модулю. Організація підтримки чат-боту набагато спрощує процес автоматизації послуг. Потенційні клієнти зайшовши на сайт кафе, можуть скористатися чат-ботом, для того щоб автоматично сформулювати заявки, наприклад, на доставку.

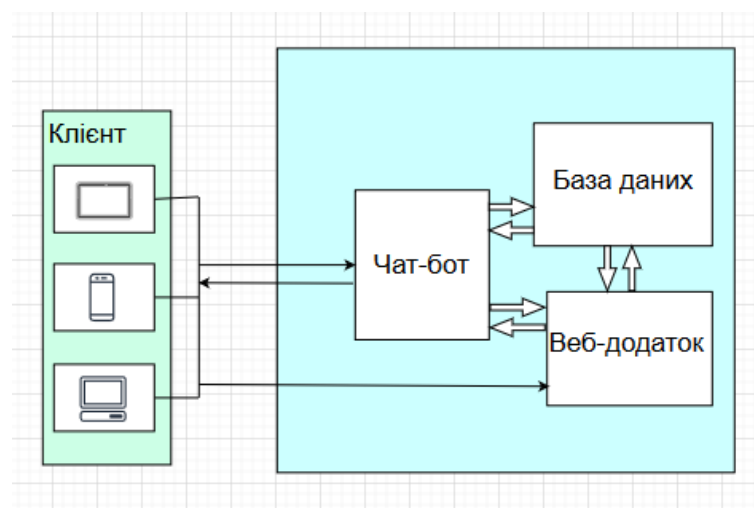


Рисунок 3.1 - Структура програмного модулю

Розробка програмного модулю передбачає проектування концепції функціонування роботи. Це надає змогу продемонструвати усі процеси створення та послідовність їх виконання, а також чітко документувати різні аспекти діяльності – дії, які необхідно зробити, способи їх здійснення, що вимагаються для цього ресурсу та ін. Таким чином, формується цілісна картина розробки.

Процес проектування складається із наступних складових: створення контекстної діаграми, її декомпозиції та варіантів використання (use-case). В першу чергу необхідно побудувати контекстну діаграму (рис. 3.2). Головним процесом моделі є розробка програмного модулю із підтримкою чат-боту.

Відповідно, на виході отримуємо вже готовий програмний продукт. Механізми представлені у вигляді розробника, що займається розробкою модулю, технічного забезпечення, мови програмування python, SQL. Усі процеси системи регулюються протоколами, які будуть використані при розробці, а також документацією API Telegram.

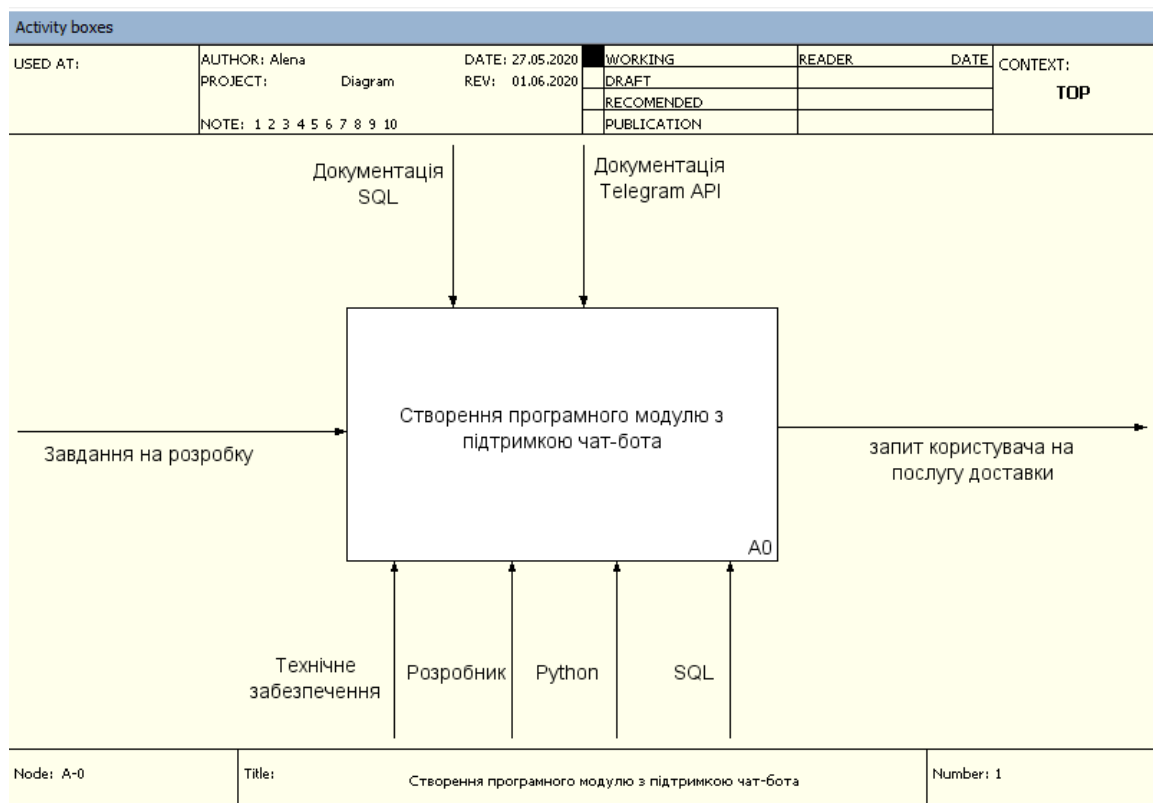


Рисунок 3.2 – Контекстна діаграма

Контекстна діаграма дає загальний опис процесу створення модулю. Для того, щоб зобразити його більш детально, була виконана декомпозиція. Розбиття головного процесу на фрагменти дозволяє ознайомитись з логікою послідовності виконання робіт для реалізації вихідного продукту. На рисунку 3.3 схематично в якості блоків показані основні процеси створення модулю. Контекстна діаграма була розбита на три підрівні (фрагменти): отримання доступу до Telegram, підключення бази даних, безпосередньо розробка модулю. Це обумовлено тим, що процес створення функціонального веб-ресурсу поділяється на три етапи:

Перший етап: надання доступу до месенджеру для здійснення маніпуляцій.

Другий етап: створення структури роботи програмного продукту.

Опис першого етапу формування даних:

- Вхідні дані: завдання на розробку.
- Вихідні дані: отримання зв'язку з Telegram.
- Управління: документація Telegram API.
- Механізми: розробник, технічне забезпечення, python.

Опис другого етапу формування даних:

- Вхідні дані: отримання зв'язку з Telegram.
- Вихідні дані: отримання зв'язку базою даних PostgreSQL.
- Управління: документація SQL.
- Механізми: розробник, python, SQL.

Опис третього етапу формування даних:

- Вхідні дані: отримання зв'язку базою даних PostgreSQL.
- Вихідні дані: запит користувача на отримання послуги доставки.
- Управління: документація Telegram API.
- Механізми: розробник, python, API Telegram.

Діаграма декомпозиції представлена на (рис.3.4).

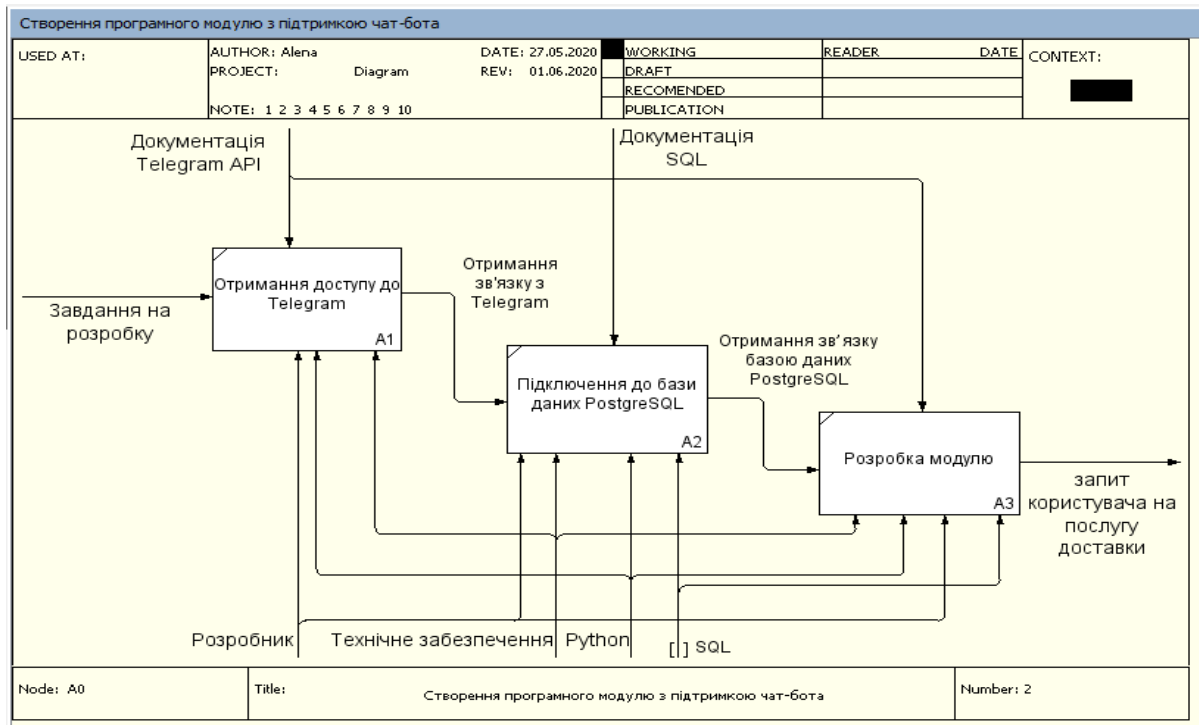


Рисунок 3.4 – Декомпозиція створення програмного модулю

Діаграма варіантів складається з графічної діаграми, яка описує дійові особи і прецеденти. У даній діаграмі спроектована система представляється у вигляді сутностей чи акторів, що взаємодіють із системою за допомогою варіантів використання. Однак, актором (actor) чи дійовою особою може бути яка завгодно сутність, що здійснює взаємодію з системою із зовні.

Отже, в межах розробленого програмного модулю взаємодіють такі актори:

- User-користувач, який може користуватись функціями бота.
- User-адмін, який може користуватись додатковими функціями бота.

У програмному модулі автоматизуються наступні процеси:

- формування заявки користувача на замовлення продуктів;
- автоматичне повідомлення менеджера про заявку;
- розсилка клієнтам повідомлення про акції та пропозиції.

Реалізуються такі функції:

- оформлення замовлення на доставку (із обраним товаром, кількості та загальної ціни, адреса доставки, контактні дані та інформацією про час);

- створення товару (для адміністратора) в смартфоні;
- створення розсилки (для адміністратора).

Після того, як визначені всі актори, які будуть взаємодіяти з системою, було сформовано перелік варіантів використання. Під час першого запуску чат-бота доступні три функції – «Меню», «Контакти», «Отзыв».

Для користувача стандартний сценарій полягає в наступному: користувач, після запуску діалогу із чат-ботом, з доступних кнопок обирає «Меню». Потреби у проходженні етапу реєстрації немає, адже чат-боту досить знати внутрішній ідентифікатор користувача в месенджері для ведення діалогу.

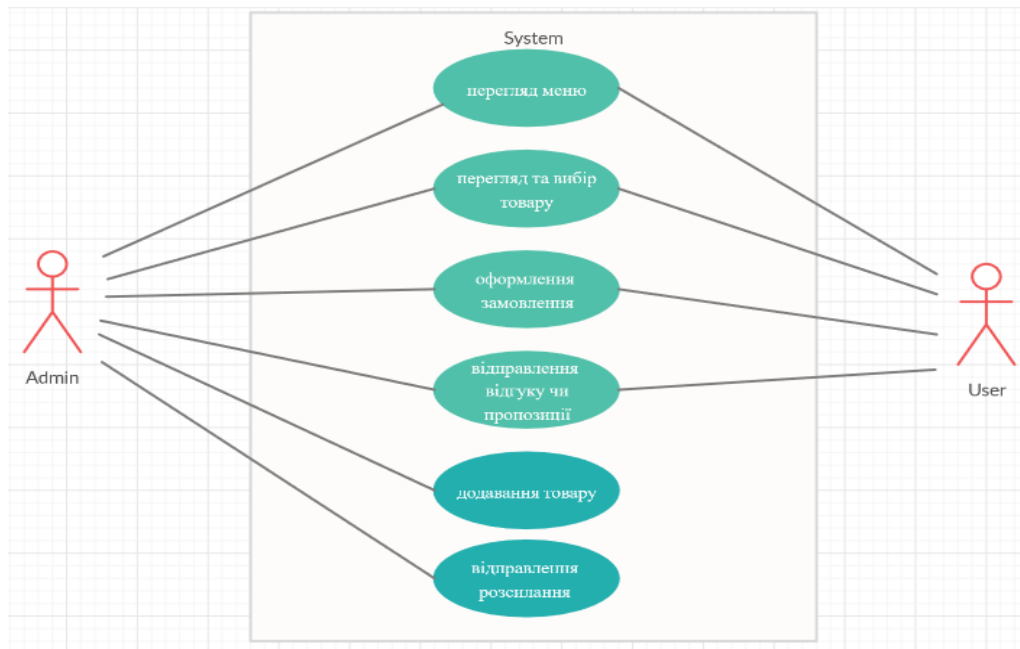


Рисунок 3.5 – Use-case діаграма

Доступні такі функції:

- перегляд меню;
- перегляд та вибір товару;
- оформлення замовлення (введення даних для покупки: ім'я, адреса, телефон);
- відправлення відгуку чи пропозиції.

Для User-адміна сценарій дещо доповнений. Він проходить у форматі вибору команд:

- можливість додавання нового товару;
- можливість оформлення розсилання для інших користувачів.

Виходячи із сформульованих даних про акторів та всіх можливих варіантів використання чат-боту, розроблена Use-Case діаграма. Вона представлена на рисунку 3.5.

На даний момент реалізовано веб-додаток кафе «Жирафф» міста Рубіжне <http://giraff.in.ua/>. Веб-додаток оформлений у вигляді лендінг пейдж, має простий та зручний інтуїтивно зрозумілий інтерфейс (рис 3.6).

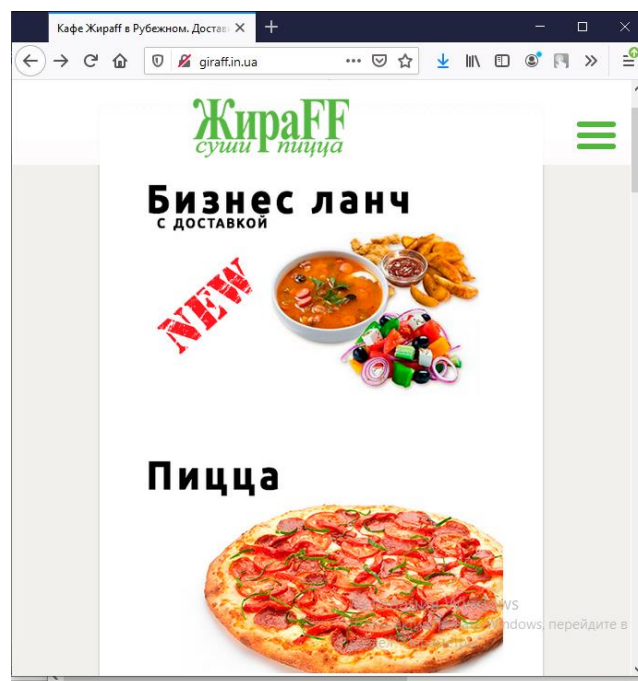


Рисунок 3.6 – Головна сторінка сайту

Веб-додаток виконує лише декілька функцій – ознайомлення клієнта з меню кафе та можливість оформити замовлення до дому. Така можливість представлена у вигляді заповнення форми «Имя» та «Телефон» и кнопки «Заказать». Через деякий час телефонує менеджер та уточнює замовлення.

Такий спосіб є не зовсім зручним для клієнтів, так як вони вимушені кожного разу чекати доки менеджер не уточнить замовлення, а це віднімає їх час. Також значним мінусом веб-додатку – є відсутність авторизації користувача, тобто задля оформлення замовлення кожного разу необхідно вводити свої дані.

Вищевикладений матеріал підтверджує необхідність створення додаткового програмного рішення для закладу громадського харчування міста Рубіжне, кафе «Жиравф», у вигляді підтримки чат-боту.

### 3.2 Структура бази даних

Для зручності роботи з БД PostgreSQL для мови Python є бібліотека gino. Спочатку потрібно установити gino простою командою:

```
$pip install gino
```

Після чого необхідно під'єднатися до бази даних:

```
await db.set_bind(f'postgresql://{db_user}:{db_pass}@{host}/gino')
```

Далі необхідно оголосити модель і імпортувати бібліотеку:

```
from gino import Gino  
db = Gino()
```

db – діє як посилання на базу даних, через яке буде проходити більшість взаємодій.

Модель – це базова концепція в Gino, це клас в Python, унаслідований від db.Model. Кожен Model підклас зіставляється з однією таблицею в базі даних. А кожен об'єкт класу представляє одну строку в таблиці. Оголосимо моделі, що мають зв'язок між собою. Таблиці Items, Category, Purchase, Users.

Таблиця Users, відповідна даної сутності, зберігає в базі даних інформацію про покупця, містить наступний набір атрибутів:

```

class User(db.Model):
    __tablename__ = 'users'
    id = Column(Integer, Sequence('user_id_seq'), primary_key=True)
    user_id = Column(BigInteger)
    language = Column(String(2))
    full_name = Column(String(100))
    username = Column(String(50))
    referral = Column(Integer)
    query: sql.Select

```

Таблиця Items бази даних містить інформацію товар: фото, ціна, інгредієнти, маса.

```

Class Item(db.Model):
    __tablename__ = 'items'
    query: sql.Select
    id = Column(Integer, Sequence('item_id_seq'), primary_key=True)
    id_category = Column(Integer)
    name = Column(String(50))
    description = Column(String(250))
    photo = Column(String(250))
    price = Column(Integer) # Ціна в копійках (потом делим на 100)

```

Головним призначенням сутності Category є угруповання товарів. Вона має наступний набір атрибутів:

```

Class Category(db.Model):
    __tablename__ = 'items'
    query: sql.Select
    id = Column(Integer, Sequence('item_id_seq'), primary_key=True)
    name = Column(String(50))
    description = Column(String(250))

```

Призначення сутності Purchase – інформування адміністраторам, зберігання необхідних даних про замовлення.

```

Class Purchase(db.Model):
    __tablename__ = 'purchases'
    query: sql.Select
    id = Column(Integer, Sequence('purchase_id_seq'), primary_key=True)
    buyer = db.Column(db.BigInteger)

```



```

item_id = Column(Integer)
amount = Column(Integer) # Цена в копейках (потом делим на 100)
quantity = Column(Integer)
purchase_time = Column(TIMESTAMP)
shipping_address = Column(JSON)
phone_number = Column(String(50))
successful = Column(Boolean, default=False)

```

Для побудови таблиць, необхідно виконати наступні команди:

```

# Create tables
db.gino: GinoSchemaVisitor
await db.gino.drop_all()
await db.gino.create_all()

```

Структура БД програмного модулю надання послуг доставки з 4 таблиць – Items, Category, Purchase, Users (рис. 3.7).

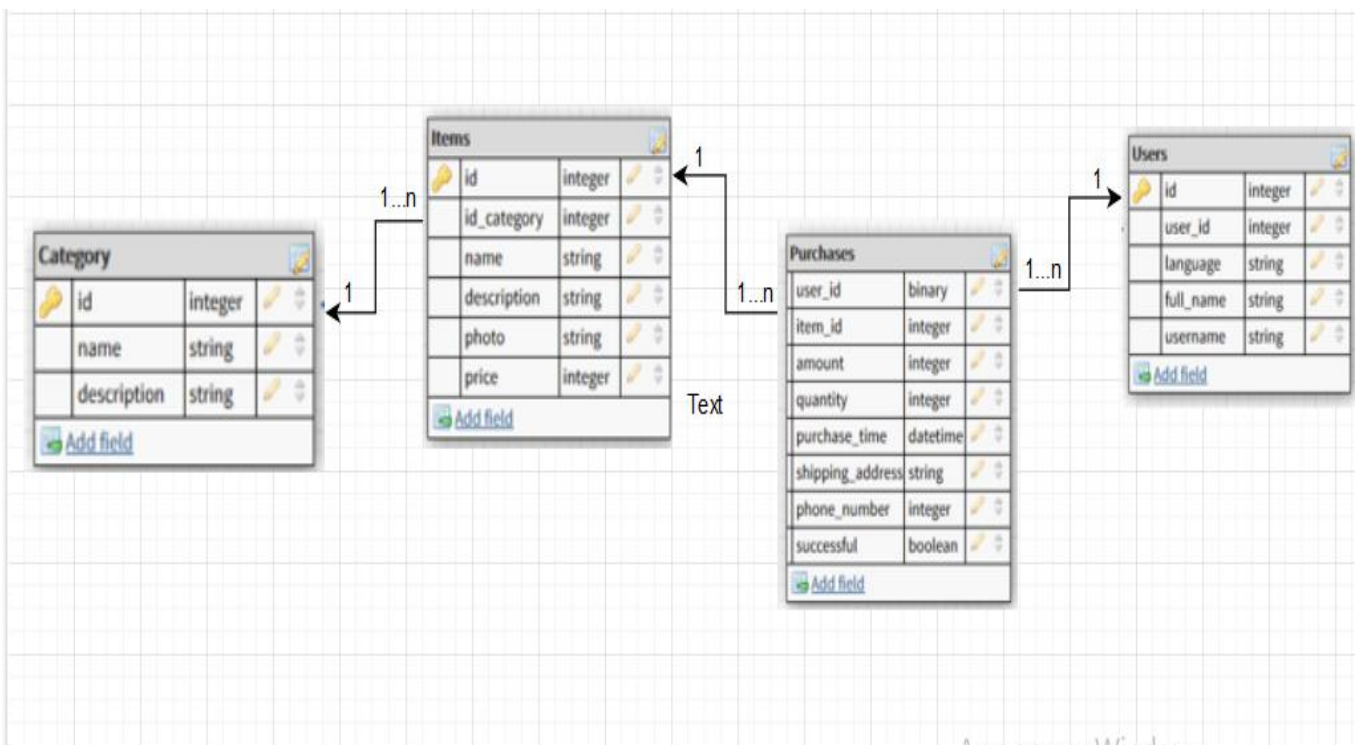


Рисунок 3.7 – Структура бази даних

### 3.3 Розробка програмного модулю

Програмний продукт передбачає версію програми для платформ Telegram. Щоб зрозуміти, яким чином буде функціонувати даний чат-бот, була розроблена схема, яка зображена на рисунку 3.8.

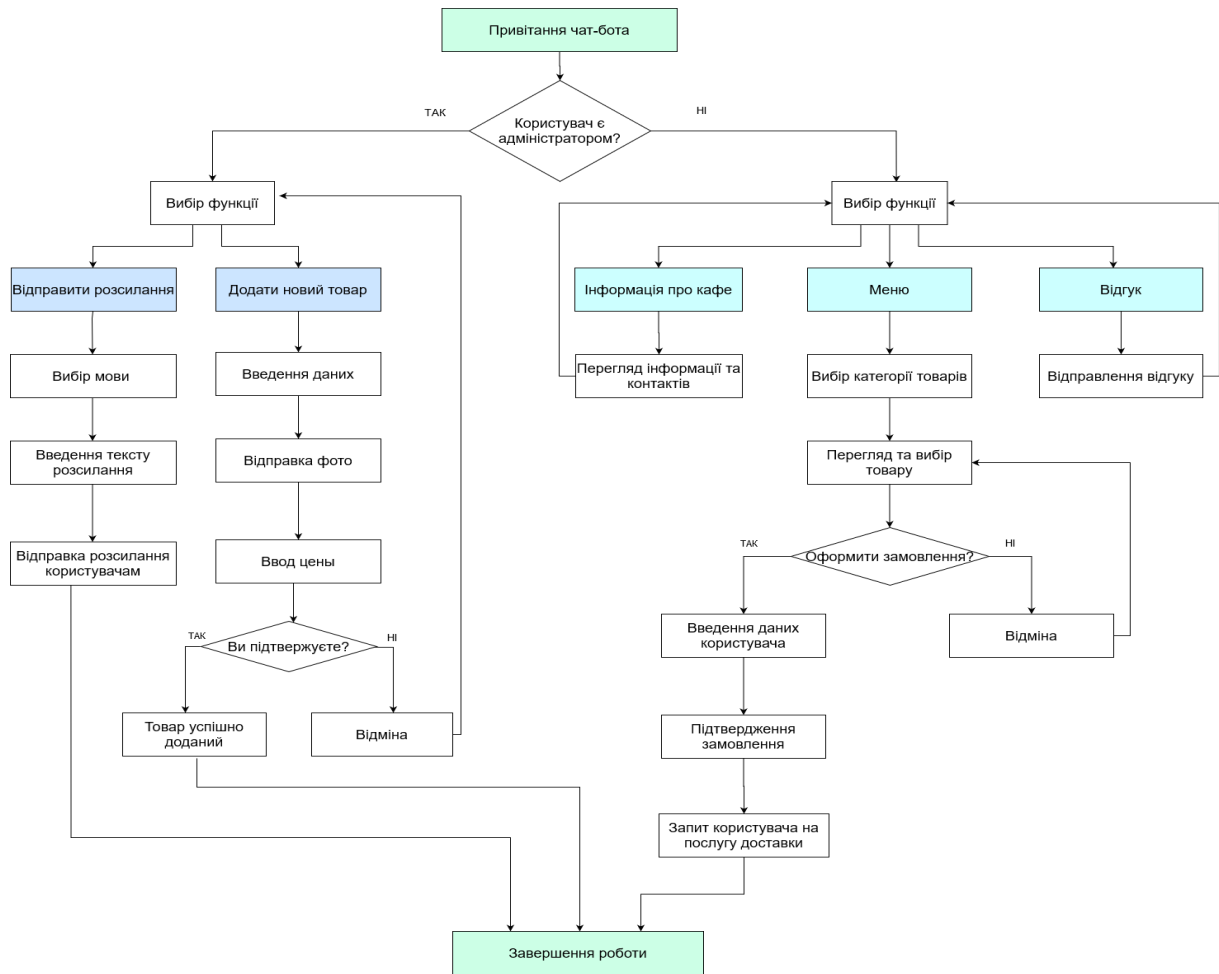


Рисунок 3.8 – Схема роботи чат – боту

Створити чат-бот можна двома способами:

1. Написання програмного коду чат-боту власноруч.
2. Створення чат-боту, використовуючи сервіси конструкторів.

Розроблений чат-бот написаний власноруч мовою Python з використанням бібліотеки aiogram. Бот по принципу дії буде використовувати на заздалегідь запрограмовані варіанти взаємодії з користувачами.

Перед тим, як приступати до розробки, необхідно зареєструвати назву боту у спеціального чат-бота «BotFather». Для реєстрації потрібно ввести команду «/newbot», після чого можна написати власне назву боту. Існує обов’язкова умова: в кінці назви має бути зазначено «Bot» або «\_bot». При дотриманні всіх умов «BotFather» видає токен (набір символів для доступу до HTTP API Telegram Bot), а також URL-адресу для доступу до чат-боту. Приклад реєстрації чат-боту представлений на рисунку 3.9.

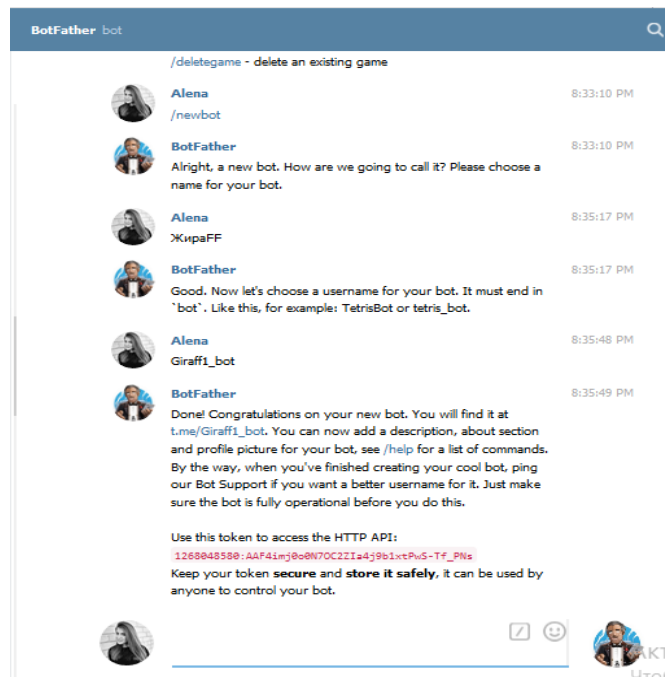


Рисунок 3.9 – Реєстрація боту

При бажанні можна установити додаткові параметри, такі як: додати іконку чат-боту, привітальне повідомлення, опис роботи боту та інше.

Після закінчення налаштувань на стороні Telegram і отримання токена можна приступити до самої розробки чат-боту.

Для початку роботи зі щойно утвореним ботом потрібно в файл `config.py` додати згенерований токен.

```
Load_dotenv()
TOKEN = os.getenv("TOKEN")
```

Щоб отримувати повідомлення від сервера Телеграм використовує polling (poll- опитувати) – постійне опитування сервера на наявність нових оновлень. Фрагмент коду для старту боту:

```
if __name__ == '__main__':
    executor.start_polling(dp, on_shutdown=on_shutdown,
on_startup=on_startup)
```

Далі можна приступати до написання коду для взаємодії з користувачем.

Користувачі Telegram можуть взаємодіяти з чат-ботами 2 способами: команди («/start», «/new» та інші) з параметрами, або кнопки. Кнопки можуть бути 2 видів: ReplyKeyboard – клавіатура кнопок, що показуються замість основною та вбудована клавіатура (inline keyboards), що кріпиться до повідомлення. У месенджері Telegram кожна клавіатура реалізована як об’єкт, а її кнопки як JSON рядок. У документації Telegram Bot API сказано, що кожна клавіатура повинна мати один обов’язковий параметр – ім’я кнопки (text), і шість необов’язкових – посилання (url), зворотні дані (callback\_data) і кнопка з можливістю покупки (pay). Приклад реалізації клавіатури одного з меню чат-боту зображений на рис. 3.10 .

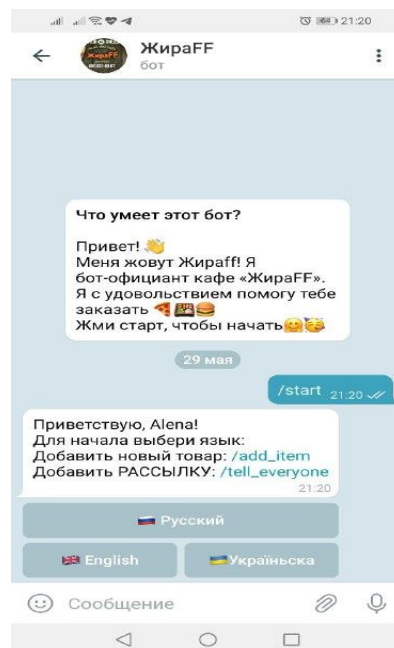


Рисунок 3.10 – Меню вибору мови

Бот має функціонал для привітання при старті розмови (дод. А). При першому запуску чат-бота користувачеві пропонується вибрати між російською та англійською мовою (рис. 3.10).

Наступним завданням була реалізація основного меню. Воно дозволяє користувачеві обрати необхідну для нього інформацію (рис. 3.11).

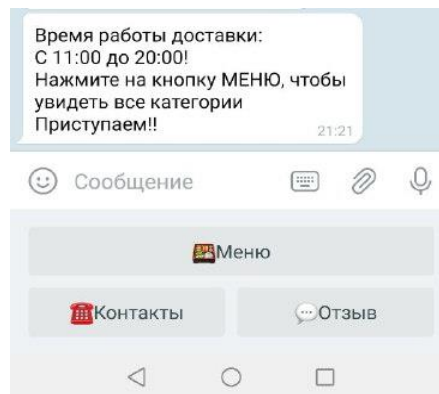


Рисунок 3.11 – Головне меню

Друга реалізована функція являє собою меню з набору категорії страв, і в залежності від вибору буде відображена інформація про страву.

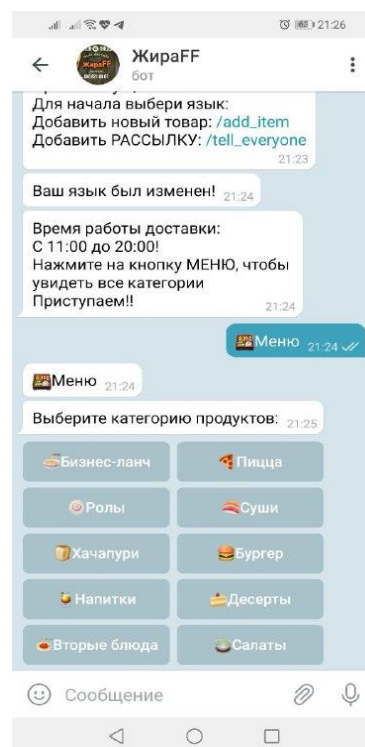


Рисунок 3.12 – Меню вибору категорії

Для відображення страв була використана наступна функція, що отримує дані з таблиці товарів:

```

async def show_items(self):
    items = await Item.query.gino.all()
    return items
# Достаем товары из базы данных
all_items = await db.show_items()
# Отправляем фотку товара с подписью и кнопкой “купить”
await call.message.answer_photo(
    photo=item.photo,
    caption=text.format(
        id=item.id,
        name=item.name,
        description=item.description,
        price=item.price / 100
    ),
    reply_markup=markup
)

```



Рисунок 3.13 – Вибір страви

Серверна частина чат-бота на платформі Telegram вирішує такі завдання:

- отримує і розпізнає повідомлення відправлене користувачем;
- отримує інформацію з бази даних шляхом виконання запитів з отриманими параметрами;
- складає і відправляє відповідь у вигляді тексту, сформованого на основі інформації користувача.

Для того щоб чат-бот мав можливість розпізнавати повідомлення користувача кожна кнопка inline-клавіатури має так звану зворотну інформацію (англ. Callback data).

Для користувачів передбачений покроковий вибір страви та оформлення замовлення. Що стосується адміністратора, то він має змогу додавати (рис. 3.14) нові страви та оформляти розсилання (рис. 3.15). Приклад реалізації наведений нижче на рисунках.

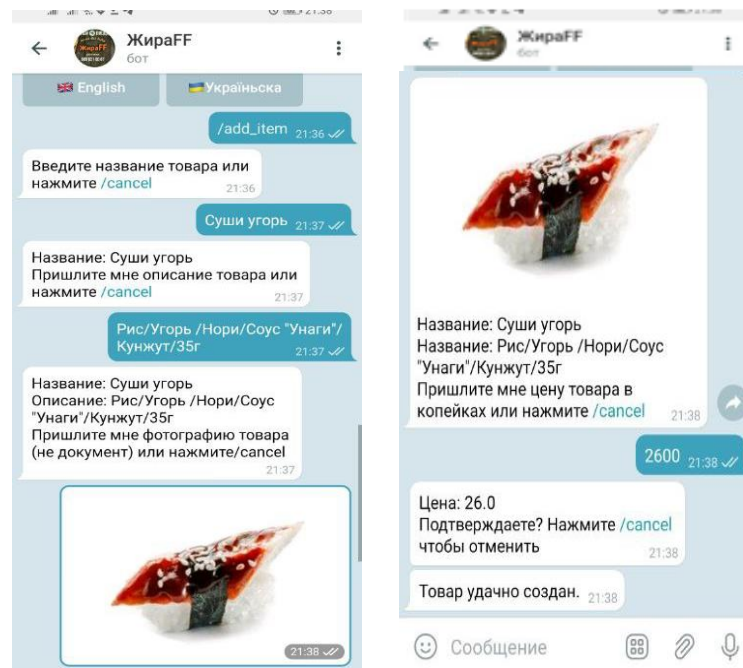


Рисунок 3.14 - Додавання нового товару

Адміністратор, натиснувши команду «/tell\_everyone», вводить та відсилає текст розсилання. Користувач в свою чергу отримує розсилання у вигляді повідомлення від чат-бота.

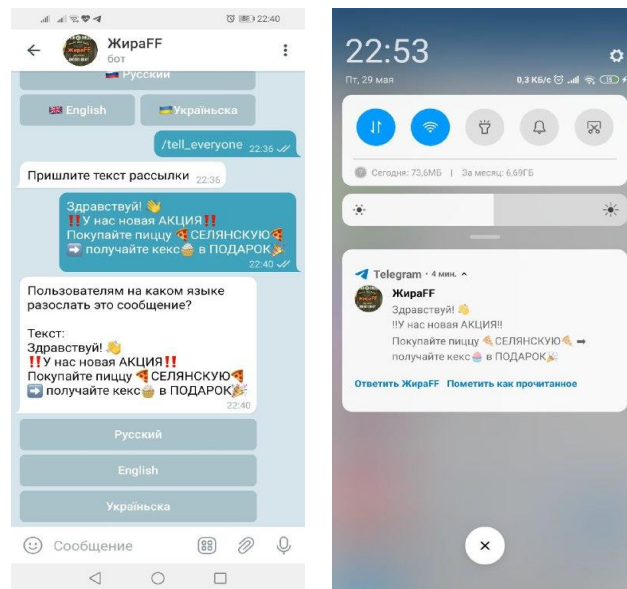


Рисунок 3.15 - Відправка та отримання розсилання

Для цих сценаріїв в практиці розробки ботів використовується підхід так названих стейт машин (State Machine), кінцевий автомат.

Ідея роботи полягає у тому, що програма може знаходитися в різноманітних станах та в кожному із них здійснювати певні дії відповідно до зазначених функцій. За такою системою необхідною умовою роботи програми є правильна визначеність всіх станів, тобто кінцевий їх набір.

```

from aiogram.dispatcher.filters.state import StatesGroup, State
class Purchase(StatesGroup):
    EnterQuantity = State()
    Approval = State()
    EnterAdress = State()
    Approval1 = State()
    EnterPhone = State()
    Approval2 = State()
classNewItem(StatesGroup):
    Name = State()
    Description = State()
    Photo = State()
    Price = State()
    Confirm = State()
classMailing(StatesGroup):
    Text = State()
    Language = State()

```



Для опису стейт машин чат-боту у проекті створений модуль `state_machine.py`

Для оформлення замовлення, користувач покроково вводить необхідну інформацію. Дані заносяться в таблицю бази даних. На рис. 3.16 наведений скрін оформлення замовлення.

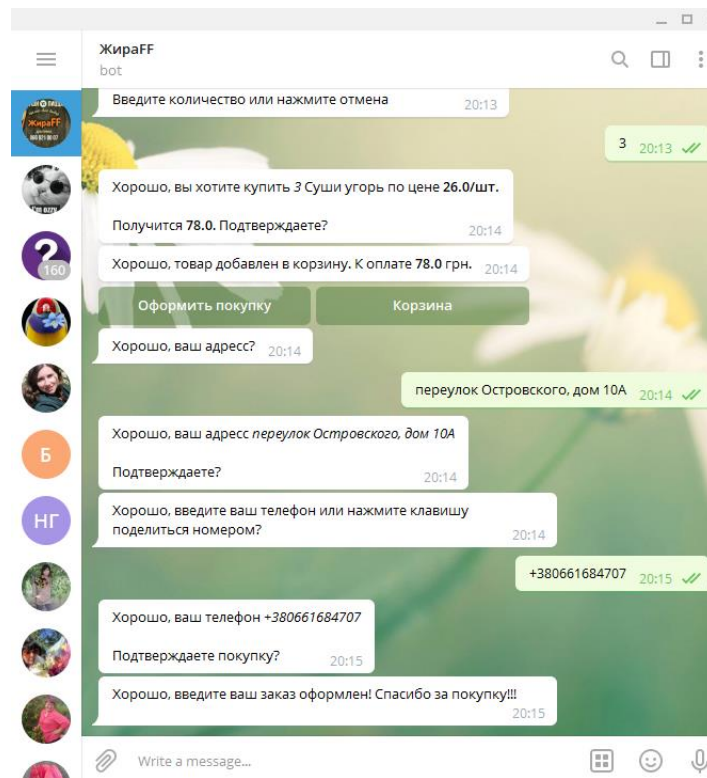


Рисунок 3.16 - Оформлення замовлення

### Висновки до розділу 3

Було розроблено програмний модуль для автоматизації надання послуг клієнтам кафе «Жираff» міста Рубіжне. Процес розробки складався з трьох етапів. На першому етапі було створено концепцію функціонування програмного модулю. Були розроблені наглядні контекстні діаграми. Це полегшило подальший процес розробки. На наступному етапі була розроблена база даних для зберігання інформації про користувача, товари та замовлення. Третій етап продемонстрував саме розробку чат-бота. Були приведені скрини його роботи, в яких продемонстровано виконання основних функцій.

## **4 ОХОРОНА ПРАЦІ**

Даний розділ присвячений аналізу обраного приміщення на відповідність нормам охорони праці і безпеки в надзвичайних ситуаціях, оскільки ці питання мають першочергові значення при виконанні дипломного проекту.

Тема дипломного проекту – програмний модуль для автоматизації надання послуг в закладі громадського харчування. В процесі проектування програмного модулю використовувалося комп'ютерне обладнання, тому аналіз потенційно небезпечних і шкідливих чинників виконується для персонального комп'ютера.

### **4.1 Загальні питання з охорони праці**

Охорона праці є невід'ємним складником умов трудової діяльності. Основні положення закріплено в Законі «Про охорону праці» [23], в якому дано їй визначення - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності. Комфортні та безпечні умови для працівника значно підвищують рівень його ефективності. Крім того, іноді вони виступають важливим фактором при виборі робочого місця, тому роботодавець має бути зацікавлений в створенні сприятливих умов та застосуванні сучасних засобів безпеки для своїх підлеглих. Працівник має право відмовитись від роботи поєднаної з небезпекою для життя або в умовах, що не відповідають нормам законодавства.

### **4.2 Аналіз стану умов праці**

Робота над створенням програмного модулю проходитиме в приміщенні приватному будинку.

Для даної роботи достатньо однієї людини, для якої надано робоче місце з персональним комп'ютером – ноутбуком.

#### 4.2.1 Вимоги до приміщення

Геометричні розміри обраного приміщення наведені в табл. 4.1.

Таблиця 4.1 – Розміри приміщення

Найменування	Значення
Довжина, м	4,15
Ширина, м	2,92
Висота, м	2,20
Площа, м <sup>2</sup>	12,118
Об'єм, м <sup>3</sup>	26,66

Згідно з ДСН 3.3.6.042-99 [27] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

Задля забезпечення потрібного рівного освітленості кімната має 2 вікна та систему загального рівномірного освітлення, що встановлена на стелі. Для дотримання визначеного рівня мікроклімату в будинку встановлено систему індивідуального опалення. Для дотримання вимог пожежної безпеки наявна ковдра та вода.

#### 4.2.2 Вимоги до організації місця праці

Організування місця праці та безпечної роботи описують опрацьовуючи наступні питання:

- ергономічні вимоги до параметрів робочого місця,
- розміщення обладнання, пристроїв та персонального комп'ютера на ньому,

- психофізіологічні особливості праці (напруженість процесу праці, режими праці та відпочинку).

Місце праці – це зона професійних дій працівника, обладнана для виконання певних операцій виробничого процесу, де взаємодіють три головні елементи праці — предмет, засоби і суб’єкт праці.

Розглянемо відповідність характеристик робочого місця нормативним. Порівняємо основні вимоги до організації робочого місця з ДСанПІН 3.3.2.007-98 [28] та відповідним фактичним значенням місця роботи (табл. 4.2).

Таблиця 4.2 – Характеристики робочого місця

Найменування параметра	Фактичне значення	Нормативне значення
Висота робочої поверхні, мм	750	680–800
Висота простору для ніг, мм	720	не менше 600
Ширина простору для ніг, мм	800	не менше 500
Глибина простору для ніг, мм	650	не менше 650
Висота поверхні сидіння, мм	430	400-500
Ширина сидіння, мм	400	не менше 400
Глибина сидіння, мм	400	не менше 400
Висота поверхні спинки, мм	370	не менше 300
Ширина опорної поверхні спинки, мм	330	не менше 300
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	700	700-800

Приміщення робочої кімнати має об’єм 26,66 м<sup>3</sup>, площу – 12,118 м<sup>2</sup>.

Температура в приміщенні протягом року коливається у межах 16–22°C, відносна вологість — близько 50%. Система вентилявання приміщення — природна неорганізована, а опалення — індивідуальне. Розміщення вікон забезпечує природне освітлення з коефіцієнтом природного освітлення не менше 1,5%, а загальне штучне освітлення, яке здійснюється за допомогою 2 LED ламп, забезпечує достатній рівень освітленості.

За показниками фактичного значення характеристик робочого місця, можна зробити висновок, що воно відповідає нормативним вимогам.

#### 4.2.3 Навантаження та напруженість процесу праці

У процесі праці людина дістає значні фізичні навантаження та нервово-емоційні напруження. Будь-який процес праці має дві складові, що пов'язані з фізичними навантаженнями на організм людини та з організуванням технологічних процесів. Першу складову розглядають з погляду енергозатрат на виконання тієї чи іншої роботи. В умовах сучасного виробництва переважають роботи помірної потужності, які за величиною енергозатрат поділяють на три категорії (табл. 4.3).

Таблиця 4.3 – Категорія робіт

Категорія робіт		Енерговитрати	
		Ват	Ккал/год
Легкі	Ia	до 139	до 120
	Iб	140–174	121–150
Середньої важкості	IIa	175–232	151–200
	IIб	233–290	201–250
Важкі		Понад 290	Понад 250

До легкої категорії належать роботи, які виконують сидячи або пов'язані з ходінням та супроводжуються незначним фізичним навантаженням.

Робота за комп'ютером відноситься до категорії легкі роботи (Ia), її виконують сидячи з періодичним ходінням. Щодо характеру організування виконання дипломної роботи, то він підпадає під нав'язаний режим, оскільки певні розділи роботи необхідно виконати у встановлені конкретні терміни.

Кожний вид професійної діяльності має свій оптимальний рівень нервово-психічної напруги, який мобілізує особу та сприяє успішній та безпечній професійній діяльності. Підвищення чи зниження цього рівня не сприяє

професійній діяльності та понижує рівень її безпечності. За ступенем нервово-психічної напруги виконання роботи можна віднести до II – III ступеня і кваліфікувати як помірно-напружений – напружений за умови успішного виконання поставлених завдань.

Під час виконання робіт використовують ПК та периферійні пристрої (лазерні та струменеві), що призводить до навантаження на окремі системи організму. Такі перекося у напруженні різних систем організму, що трапляються під час роботи з ПК, зокрема, значна напруженість зорового аналізатора і довготривале малорухоме положення перед екраном, не тільки не зменшують загального напруження, а навпаки, призводять до його посилення і появи стресових реакцій.

З метою запобігання перевантаження організму як в цілому, так і окремих його функціональних систем, передусім зорового та рухового аналізаторів, центральної нервової системи, необхідно передбачити обмеження загального часу щоденної роботи з ПК – до 8 годин. Також необхідно встановити додаткові перерви, для розробників програм - тривалістю 15 хв через кожен годину роботи.

### **4.3 Виробнича санітарія**

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

#### **4.3.1 Аналіз небезпечних шкідливих факторів при виробництві (експлуатації) виробу**

Роботу, пов'язану з ЕОП з ВДТ, у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і ПП, виконують із забезпеченням виконання НПАОП

0.00-7.15-18 [31] «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», які встановлюють вимоги безпеки до обладнання робочих місць, до роботи із застосуванням ЕОМ з ВДТ і ПП. Переважно роботи за проектами виконують у кабінетах чи інших приміщеннях, де використовують різноманітне електрообладнання, зокрема персональні комп'ютери (ПК) та периферійні пристрої.

Основними робочими характеристиками персонального комп'ютера є наступні:

- робоча напруга  $U = +220\text{В} \pm 5\%$ ;
- робочий струм  $I = 2\text{А}$ ;
- споживана потужність  $P = 350\text{Вт}$ .

Робоче місце має відповідати вимогам Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затверджених постановою Головного державного санітарного лікаря України від 10.12.98 N 7 [28].

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 4.4).

Таблиця 4.4 – Аналіз небезпечних і шкідливих виробничих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Кількісна оцінка	Нормативні документи
1	2	3	4
<b>Фізичні фактори</b>			
- недостатність природного світла	Порушення умов праці (вимог до праці -відсутність вікон у приміщенні)	2	[30]
- недостатнє освітлення робочої зони	Порушення гігієнічних параметрів виробничого середовища	3	[30]
- підвищений рівень напруги електричної мережі, замикання якої може відбутися через тіло людини	Порушення умов праці (вимог до приміщення)	4	[29]
<b>Психофізіологічні</b>			

Продовження табл. 4.4			
1	2	3	4
- нервово-психічна перевантаження (розумове, перенапруження аналізаторів-зорових)	- пошук інформації для постановки теми; - пошук та аналіз аналогів і літератури; - пошук наявних технологій, моделювання та аналіз алгоритмів; - виконання роботи за темою диплома, тестування; - оформлення роботи	7	[31] [28]
- фізичні (статичне – сидіння)	порушення умов праці (організації місця праці- сидіння користувача, ) та організації робочого часу - безперервна робота)	2	[31] [28]

### 4.3.2 Пожежна безпека

Пожежа — це неконтрольоване горіння, яке супроводжується виділенням тепла, світла, диму та інших продуктів. Горіння виникає за таких трьох умов: наявності окисника, наявності горючої речовини, наявності температури, за якої горюча речовина може самостійно горіти. Якщо немає хоча б однієї із цих умов, горіння стає неможливим. На цьому постулаті ґрунтується переважна більшість профілактичних заходів, спрямованих на відвернення пожеж.

Причинами пожеж електроустаткування може бути коротке замикання між струмоведучими частинами обладнання (шини, електроди), перевантаження напруги, пошкодження обладнання та електропроводки.

Дане приміщення, площею 12,118 м<sup>2</sup>, згідно ДСТУ Б В.1.1-36:2016 [32], відноситься до категорії "В" (пожежонебезпечної).

Горючими матеріалами в приміщенні, де розташовані ЕОМ, є:

1) поліамід – матеріал корпусу мікросхем, горюча речовина, температура самозаймання 420° С,

2) полівінілхлорид – ізоляційний матеріал, горюча речовина, температура запалювання 335° С, температура самозаймання 530° С,



3) склотекстоліт ДЦ – матеріал друкарських плат, важкогорючий матеріал, показник горючості 1.7А, не схильний до температурного самозаймання,

4) пластикат кабельний №.489 – матеріал ізоляції кабелів, горючий матеріал, показник горючості більше 2.1,

5) деревина – будівельний і обробний матеріал, з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, температура запалювання 255° С, температура самозаймання 399° С.

6) синтетичні тканини – легкозаймистий матеріал.

Продуктами згорання, що виділяються на пожежі, є: окис вуглецю; сірчистий газ; окис азоту; синильна кислота; акромін; фосген; хлор і ін. При горінні пластмас, окрім звичних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол.

Переважна більшість пожеж починається із невеличкого вогнища. Тому його своєчасну ліквідацію розглядаємо як профілактичний захід щодо недопущення його розширення до масштабів пожежі. Ліквідувати вогнище можна, усунувши одну із трьох умов виникнення горіння. Видалити горючу речовину із вогнища не завжди можна, а припинити доступ кисню до неї або/і понизити її температуру можна завжди, якщо своєчасно використати первинні засоби гасіння пожеж: воду, пісок або вогнегасники.

Для дотримання вимог пожежної безпеки наявна вода, ковдра та пісок (на вулиці).

Вода — універсальний засіб для гасіння пожеж, оскільки її застосування завдяки випаровуванню дає змогу як понизити температуру горючої речовини, так і зменшити доступ кисню до неї. Проте нею не можна гасити електроустановки під напругою та легкозаймисті рідини. Для цього треба використовувати пісок, хоча він є менш ефективним.

## 4.4 Гігієнічні вимоги до параметрів виробничого середовища

### 4.4.1 Освітлення

Освітлення приміщень та місць праці належить до тих чинників, від якого суттєво залежать працездатність і здоров'я працівників. Світло регулює всі функції людського організму і впливає на психологічний стан і настрій, обмін речовин, гормональний фон і розумову активність.

Найздоровіше освітлення забезпечує природне світло. При цьому, найбільш ефективно його використання можливе, якщо глибина приміщень не перевищує 6 м.

Окрім того, хорошим вирішенням можуть бути скляні перегородки, що забезпечують зорову і звукову ізоляцію, але в той же час не перешкоджають проникненню природного світла. У приміщенні, де розташовані ЕОМ передбачається природне бічне освітлення, рівень якого відповідає ДБН В.2.5-28:2018 [29]. Для штучного освітлення абсолютна освітленість в площині функціональної поверхні має дорівнювати 300–500 Лк.

Розрахунок природного освітлення. Для будівель виробництв світловий коефіцієнт приймається в межах не менше 1/6-1/10:

$$\sqrt{a^2 + b^2} * S_b = \left(\frac{1}{8} \div \frac{1}{10}\right) * S_n \quad (4.1)$$

де  $S_b$  – площа віконних прорізів,  $m^2$ ;  $S_n$  – площа підлоги,  $m^2$ .

$$S_n = a * b = 4,15 * 2,92 = 12,118 \text{ } m^2$$

$$S_{в і к} = \frac{1}{8} * 12,118 = 0,125 * 12,118 = 1,51 \text{ } m^2$$

Приймаємо 2 вікна площею 1,22  $m^2$  кожне.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для

створення заданої освітленості при загальному рівномірному освітленні. Розрахунок кількості світильників п виробляється по формулі (4.2):

$$n = \frac{E \cdot S \cdot Z \cdot K}{F \cdot U \cdot M} \quad (4.2)$$

де  $E$  – нормована освітленість робочої поверхні, визначається нормами – 300 лк;  $S$  – освітлювана площа,  $m^2$ ;  $S = 12,118 m^2$ ;  $Z$  – поправочний коефіцієнт світильника ( $Z = 1,15$  для ламп розжарювання та ДРЛ;  $Z = 1,1$  для люмінесцентних ламп) приймаємо рівним 1,1;  $K$  – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;  $U$  – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575  $M$  – число ламп в світильнику – 2;  $F$  – світловий потік лампи – 1200лм (для ЛБ-12Вт).

Підставивши числові значення у формулу (4.2), отримуємо:

$$n = \frac{300 \cdot 12,118 \cdot 1,1 \cdot 1,5}{1200 \cdot 0,575 \cdot 2} = \frac{5998}{1380} = 4,3$$

За результатами розрахунку приймаємо кількість освітлювальних установок - 4шт.

#### 4.4.2 Шум та вібрація

Також мають значний вплив на діяльність такі чинники як шум та вібрація. Шум як стрес-чинник є загальнобіологічним подразником, який негативно впливає на всі органи і системи організму. У разі тривалого систематичного впливу шуму може виникнути патологія з переважним ураженням слуху, центральної нервової і серцево- судинної систем. Допустимі рівні звукового тиску у октавних смугах частот, еквівалентні рівням звуку на місцях праці, що

враховують характер роботи та умови праці регламентовано ДСН 3.3.6.037-99 [25], витяг з якого подано в табл. 4.3.

Таблиця 4.5 – Допустимі рівні звукового тиску / шуму на робочих місцях

Робочі місця	Рівні звукового тиску, дБ в октавних смугах із середньгеометричними частотами, Гц								Рівень звуку, дБА
	63	125	250	500	1000	2000	4000	8000	
1	2	3	4	5	6	7	8	9	10
І. Творча діяльність, керівна робота з підвищеними вимогами, наукова діяльність, конструювання та проектування, програмування, викладання та навчання, лікарська діяльність; робочі місця у приміщеннях - дирекції, проектно-конструкторських бюро, розраховувачів, програмістів обчислювальних машин у лабораторіях для теоретичних робіт та обробки даних, прийому хворих у медпунктах	71	61	54	49	45	42	40	38	50

Рівень шуму, що супроводжує роботу програмістів за персональними комп'ютерами, а також зовнішніми чинниками, коливається у межах 50–65 дБА - за нормами ДСН 3.3.6.037-99 [25]. Шум такої інтенсивності на тлі високого ступеня напруженості праці негативно впливає на функціональний стан користувачів. Тому рекомендують знижувати фактичний рівень шуму у приміщеннях, де створюють комп'ютерні програми, 40 дБА, а в приміщеннях, де виконують роботу, що потребує зосередженості, — до 55 дБА. У залах опрацювання інформації та комп'ютерного набору рівні шуму не повинні перевищувати 65дБА.

Існують рекомендації щодо пониження шуму. Так на шляху поширення шуму в приміщенні розміщуються штучні поглиначі, стелю або стіни облицьовують звукопоглинальним матеріалом, використовують фіранки.

У даному приміщенні вимірний рівень звукового шуму коливається від 40 до 46 дБА. Тому, так як рівень шуму не перевищує гранично допустимих величин, заходи для зниження шуму не проводяться.

Вібрація на робочому місці в приміщенні, що розглядається, відповідає нормам ДСН 3.3.6.039-99 [26].

Допустимий рівень вібрацій на робочому місці:

- для 1 ступеня шкідливості до 3 дБ;
- для 2-3 -1-6 дБ;
- для 3 -більше 6 дБ.

Для захисту від електромагнітного випромінювання передбачаються наступні заходи:

- застосування нових плазмових моніторів,
- збільшення відстані робочого місця не менше, ніж на 0,5 м від джерел електромагнітного випромінювання, оскільки напруженість електричного поля зменшується при віддаленні від джерела поля;
- встановлення раціональних режимів роботи персоналу (обмеження часу перебування);
- раціональне розміщення в робочому приміщенні устаткування, що випромінює електромагнітну енергію.

#### **Висновки до розділу 4**

У даному розділі були розглянуті умови праці за робочим місцем під час виконання дипломної роботи. Розміри приміщення та параметри робочого місця відповідають нормам ДСанПІН 3.3.2.007-98 [28] з охорони праці. Було проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити виробничу санітарію. Наприклад, задля забезпечення достатнього освітлення необхідно щоб в приміщенні були розташовані 2 вікна площею 1,22 м<sup>2</sup> кожне для природного освітлення та 4 світильниками - для штучного. Також у приміщенні необхідно щодня робити вологе прибирання та провітрювати його. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації щодо поліпшення умов праці. При інтенсивній роботі під час виконання дипломного проекту рекомендовано щогодини робити перерву на 15 хвилин. Також рекомендується робити вправи для очей для запобігання проблем із зором.

## ВИСНОВКИ

У наш час дуже стрімко набирають популярність заклади громадського харчування, і, як наслідок, зростає потреба у впровадженні нових технологій задля отримання переваг над конкурентами. У вік значного розвитку комп'ютерних технологій впровадження інформаційно-технологічних інновацій є необхідністю. Вони дають можливість задовольнити найкращим способом потреби сучасних людей. А так як, сьогодні темп життя є стрімким, то потреби людей зосередженні на економії дорогоцінного часу та підвищенні зручності отримання необхідних послуг. Все більше людей надають перевагу послугі доставки їжі, як додому, так і в офіс на роботу. Для цього виникає необхідність у впровадженні електронного меню. А також, сервісів, де його можна знайти.

У зв'язку із сьогоднішньою ситуацією із розповсюдженням епідемії COVID-19, більшість країн світу знаходиться на карантині. У тому числі і Україна. Більшість послуг, які надають заклади громадського харчування є обмеженими. До послаблення карантину основною послугою закладів – була послуга доставки. Тому актуальність автоматизації надання послуг у закладах громадського харчування є очевидною.

В рамках реалізації дипломного проекту були виконані поставлені завдання. Був проведений аналіз актуальності теми розробки. Проведений аналіз існуючих сервісів показав, що створення програмного модулю з підтримкою чат-боту є найкращим сучасним рішенням автоматизації надання послуг в закладах громадського харчування.

Під час реалізації програмного модулю було створено графічне представлення у вигляді структурної схеми, контекстних діаграм, що показують порядок виконання робіт та діаграми варіантів використання. Були описані можливі сценарії користувачів (адміністратора та звичайного користувача). Це полегшило подальшу розробку модулю. Для розробки була обрана мова Python та бібліотека aiogram, для зручної роботи з базами даних - СУБД PostgreSQL, для зв'язку з Telegram використовувався Telegram Bot API.

Результатом виконання дипломного проекту є програмний модуль, що автоматизує процес надання послуг у закладі громадського харчування – кафе міста Рубіжне «ЖираFF». За рахунок підтримки чат-бота автоматизуються такі процеси:

- формування заявки користувача на замовлення доставки продуктів;
- автоматичне повідомлення менеджера про заявку;
- розсилка клієнтам повідомлення про акції та нові пропозиції.

Окрім того, реалізована можливість для адміністратора додавати нові товари в телефонному режимі, відправляти нові розсилання користувачам та відповідати на повідомлення зворотного зв'язку з клієнтами.

Даний проект є корисним як для клієнтів кафе, так і для його власників. Використовуючи чат-бот, клієнти зможуть зручно та швидко обрати необхідний товар та відправити заявку на доставку із свого мобільного де б вони не знаходилися. Це, безумовно, вплине на розширення клієнтської бази кафе і, як наслідок, збільшення прибутку.

Також перевагою створеного модулю є безкоштовність та доступність для будь-якого користувача, у якого є встановлений месенджер Телеграм.

Були розглянуті та описані норми охорони праці та безпеки в надзвичайних ситуаціях.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Аналіз ринку громадського харчування (HoReCa, кейтеринг, а також харчування на АЗС). URL: <https://pro-consulting.ua/ua/pressroom/analiz-rynka-obshestvennogo-pitaniya-horeca-kejtering-takzhe-pitanie-na-azs> (дата звернення 12.05.20).
2. Організація обслуговування у закладах ресторанного господарства : Підручник / За ред. П'ятницької Н. О. – 2-ге вид. перероб. та допов.– К. : Центр учбової літератури, 2011 – 584 с.
3. Діджиталізація: найкращі цифрові рішення для HoReCa. URL: <https://blog.metro.ua/didzhytalizatsiya-najkrashhi-tsyfrovi-rishennya-dlya-horeca/> (дата звернення 12.05.20).
4. Котельникова А.В. Обеспечение конкурентоспособности предприятия общественного пи-тания: Автореф. дис. канд. экон. Наук. -Уфа, 2009. -25с.
5. Рубцова, Н.В. (2020). Инновации в индустрии гостеприимства. Russian Journal of Innovation Economics. 10. 457-466. 10.18334/vines.10.1.100664.
6. УДК 004 : 005.936.3 - 022.51:62-933.6 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ АВТОМАТИЗАЦІЇ МАЛОГО БІЗНЕСУ Дорошук М.О., бакалавр; Трішин Ф.А., к.т.н., доцент; Жигайло О.М., к.т.н. Одеська національна академія харчових технологій, м. Одеса.
7. УДК 304.2: ДОДАТКОВІ ПОСЛУГИ Й ІННОВАЦІЇ В СОЦІОКУЛЬТУРНОМУ ПРОСТОРІ. О. Є. Завгородня, аспірант, М. М. Ткач, кандидат історичних наук, професор. Київський національний університет культури і мистецтв, м. Київ.
8. Mamuladze, Gela. (2018). РОЛЬ ИНТЕРНЕТ - ТЕХНОЛОГИИ В РАЗВИТИИ ЭЛЕКТРОННОЙ ТОРГОВЛИ: ТЕНДЕНЦИИ И ПЕРСПЕКТИВЫ.
9. Бацына Я.В. ИСПОЛЬЗОВАНИЕ И ПЕРСПЕКТИВЫ ЦИФРОВЫХ ТЕХНОЛОГИЙ В РЕСТОРАННОМ БИЗНЕСЕ // Вестник Алтайской академии



экономики и права. – 2019. – № 4-1. – С. 10-18;  
URL: <http://vaael.ru/ru/article/view?id=399> (дата обращения: 13.05.2020).

10. Электронное меню для ресторана. URL:  
(<https://tipymenu.com/ru/articles/forowners>) (дата звернення 12.05.20).

11. Аналіз ринку доставки продуктів харчування України. 2019 рік. URL:  
<https://pro-consulting.ua/ua/issledovanie-rynka/analiz-rynka-dostavki-produktov-ukrainy-2018-god> (дата звернення 12.05.20).

12. Богуславский А.Л. Маркетинговые коммуникации в сфере общественного питания // Маркетинговые коммуникации. — 2020. — No1. — С.20–27.

13. Оліфіров О. В. Інформаційні технології у готельному і ресторанному бізнесі / О. В. Оліфіров, А. П. Лутай. – Донецьк : ДонНУЕТ, 2011. – 235 с.

14. <http://pervak.kiev.ua/>

15. <http://gs.statcounter.com/social-media-stats>

16. Як змінилась статистика соціальних мереж в Україні до 2018? URL:  
<https://marketer.ua/ua/social-network-of-ukraine-how-did-the-statistics-change-until-2018/> (дата звернення 10.05.20).

17. Двадцать главных трендов digital-маркетинга 2020 года. URL:  
<https://vc.ru/marketing/102519-dvadcat-glavnyh-trendov-digital-marketinga-2020-goda> (дата звернення 13.05.20).

18. Vorontsova, Liliya & Timiryasova, Asiya. (2012). Автоматизированная информационная система учета заказов и списания продуктов для пунктов общественного питания.

19. Петрук Ю. Впровадження інформаційних та автоматизованих технологій в діяльність підприємств ресторанного господарства в сучасних умовах / Ю. Петрук // Вісник КТУЕП. – 2012. – №1. – С. 167-172.

20. П'ятницька Г. Інноваційні ресторани технології: основи теорії : навч. посіб. для вищ. навч. закл. / Г. П'ятницька, Н. П'ятницька. — Київ : КондорВиробництво, 2013. — 250 с.

21. python-telegram-bot [Електронний ресурс] – Режим доступу: <https://github.com/python-telegram-bot/python-telegram-bot>
22. Лутц М. Программирование на Python, том II, 4-е издание. – Пер. сангл. – СПб.: Символ-Плюс, 2011. – 992 с.
23. Закон України "Про охорону праці". Вводиться в дію Постановою ВР № 2695-ХІІ від 14.10.92, ВВР, 1992, № 49, ст.669. - Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/2694-12](http://www.url:https://zakon.rada.gov.ua/laws/show/2694-12)
24. Кодекс законів про працю України. Затверджується Законом № 322-VIII від 10.12.71 ВВР, 1971. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/322-08](http://www.url:https://zakon.rada.gov.ua/laws/show/322-08)
25. ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку
26. ДСН 3.3.6.039-99 Санітарні норми виробничої загальної та локальної вібрації
27. Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. Постанова N 42 від 01.12.99. Режим доступу: [www. URL: https://zakon.rada.gov.ua/rada/show/va042282-99](http://www.url:https://zakon.rada.gov.ua/rada/show/va042282-99)
28. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПН 3.3.2.007-98. Затверджено Постановою Головного державного санітарного лікаря України 10 грудня 1998 р. N 7. Режим доступу: [www. URL: https://zakon.rada.gov.ua/rada/show/v0007282-98](http://www.url:https://zakon.rada.gov.ua/rada/show/v0007282-98)
29. Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом. Наказ від 1 липня 2016 року N 204. Режим доступу: [www. URL: http://epicentre.co.ua/dstu/doc28522.html](http://www.url:http://epicentre.co.ua/dstu/doc28522.html)
30. ДБН В.2.5-28:2018 «Природне і штучне освітлення». Режим доступу: [www. URL: http://www.minregion.gov.ua/wp-content/uploads/2018/12/V2528-1.pdf](http://www.minregion.gov.ua/wp-content/uploads/2018/12/V2528-1.pdf)
31. НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». Зареєстровано в

Міністерстві юстиції України 25 квітня 2018 р. за № 508/31960. Режим доступу:  
www. URL: <https://zakon.rada.gov.ua/laws/show/z0508-18>

32. ДСТУ Б В.1.1-36:2016 «Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою». Наказ від 15.06.2016 №158. Режим доступу: www. URL: <https://zakon.rada.gov.ua/rada/show/v01588>

## Додаток А

### Лістинг коду `handlers.py`

```

1  from asyncio import sleep
2  import datetime
3  import asyncio
4  import datetime
5  import emoji
6  from aiogram import types
7  from aiogram.dispatcher import FSMContext
8  from aiogram.dispatcher.filters import CommandStart
9  from aiogram.types import (Message, InlineKeyboardMarkup, InlineKeyboardButton,
10                             ReplyKeyboardMarkup, KeyboardButton,
11                             ReplyKeyboardRemove, CallbackQuery, LabeledPrice, PreCheckoutQuery)
12  from aiogram.utils.callback_data import CallbackData
13
14  import database
15  import states
16  from config import lp_token, admin_id
17  from load_all import dp, bot, _
18
19  db = database.DBCommands()
20
21  buy_item = CallbackData("buy", "item_id")
22  buyer_id = CallbackData("buyer", "buyer_id")
23
24  # Для команды /start есть специальный фильтр, который можно тут использовать
25  @dp.message_handler(CommandStart())
26  async def register_user(message: types.Message):
27      chat_id = message.from_user.id
28      referral = message.get_args()
29      user = await db.add_new_user(referral=referral)
30      id = user.id
31      count_users = await db.count_users()
32
33      # Отдадим пользователю клавиатуру с выбором языков
34      languages_markup = types.InlineKeyboardMarkup(
35          inline_keyboard=
36          [
37              [
38                  types.InlineKeyboardButton(text=emoji.emojize(':Russia: Русский',
39 use_aliases=True), callback_data="lang_ru")],
40              [
41                  types.InlineKeyboardButton(text=emoji.emojize(':United_Kingdom:
42 English', use_aliases=True), callback_data="lang_en"),
43                  types.InlineKeyboardButton(text=emoji.emojize(':Ukraine: Українська',
44 use_aliases=True), callback_data="lang_uk"),
45              ]])
46
47      bot_username = (await bot.me).username
48      bot_link = f"https://t.me/{bot_username}?start={id}"
49
50      text = _("Приветствую, {user_username}!\n"
51              "Для начала выбери язык:"
52              ).format(
53          user_username=message.from_user.first_name,
54      )
55
56      if message.from_user.id == admin_id:
57          text += _("\n"

```

```

58         "Добавить новый товар: /add_item"
59         "\n"
60         "Добавить РАССЫЛКУ: /tell_everyone"
61     )
62     await message.answer(text, reply_markup=languages_markup)
63 @dp.callback_query_handler(text_contains="lang")
64 async def change_language(call: CallbackQuery):
65     await call.message.edit_reply_markup()
66
67     first_menu_markup = types.ReplyKeyboardMarkup(
68         keyboard=
69         [
70             [
71                 types.KeyboardButton(text="Меню")
72             ],
73             [
74                 types.KeyboardButton(text=emoji.emojize(':phone:Контакты',
75 use_aliases=True)),
76                 types.KeyboardButton(text=emoji.emojize(':speech_balloon:Отзыв',
77 use_aliases=True)),
78             ],
79         ], resize_keyboard=True, one_time_keyboard=True
80     )
81
82     lang = call.data[-2:]
83     await db.set_language(lang)
84     text1 = _(
85         # "Сейчас в базе {count_users} человек!\n"
86         "Время работы доставки:"      "\n"
87         "С 11:00 до 20:00!"              "\n"
88         "Нажмите на кнопку МЕНЮ, чтобы увидеть все категории"  "\n"
89         "Приступаем!!"
90     )
91     # После того, как мы поменяли язык, в этой функции все еще указан старый,
92     поэтому передаем locale=lang
93     await call.message.answer(_("Ваш язык был изменен!", locale=lang))
94     await call.message.answer(text1, reply_markup=first_menu_markup)
95
96     second_menu_markup = types.ReplyKeyboardMarkup(
97         keyboard=
98         [
99             [
100                types.KeyboardButton(text="На главную"),
101                types.KeyboardButton(text="Меню"),
102                types.KeyboardButton(text="Корзина"),
103            ]
104        ],
105        resize_keyboard=True,
106        one_time_keyboard=True
107    )
108
109     # Для кнопки МЕНЮ
110 @dp.message_handler(text="Меню")
111 async def menu(message: types.Message):
112     # Отдадим пользователю клавиатуру с выбором языков
113     await message.answer(_("МЕНЮ:"), reply_markup=second_menu_markup)
114     menu_markup = types.InlineKeyboardMarkup(
115         inline_keyboard=
116         [
117             [
118                 types.InlineKeyboardButton(text=emoji.emojize(':ramen:Бизнес-
119 ланч', use_aliases=True), callback_data="launch"),
120                 types.InlineKeyboardButton(text=emoji.emojize(':pizza:Пицца',
121 use_aliases=True), callback_data="pizza")

```

```

122         ],
123         [
124             types.InlineKeyboardButton(text=emoji.emojize(':fish_cake:Ролы',
125 use_aliases=True), callback_data="role"),
126             types.InlineKeyboardButton(text=emoji.emojize(':sushi:Суши',
127 use_aliases=True), callback_data="sushi")
128         ],
129         [
130             types.InlineKeyboardButton(text=emoji.emojize(':bread:Хачапури',
131 use_aliases=True), callback_data="hachapur"),
132
133             types.InlineKeyboardButton(text=emoji.emojize(':hamburger:Бургер',
134 use_aliases=True), callback_data="burger")
135         ],
136         [
137
138             types.InlineKeyboardButton(text=emoji.emojize(':tropical_drink:Напитки',
139 use_aliases=True), callback_data="juse"),
140             types.InlineKeyboardButton(text=emoji.emojize(':cake:Десерты',
141 use_aliases=True), callback_data="desert")
142         ],
143         [
144             types.InlineKeyboardButton(text=emoji.emojize(':spaghetti:Вторые
145 блюда', use_aliases=True), callback_data="bluda"),
146             types.InlineKeyboardButton(text=emoji.emojize(':tea:Салаты',
147 use_aliases=True), callback_data="salat")
148         ]
149     ]
150 )
151 bot_username = (await bot.me).username
152 bot_link = f"https://t.me/{bot_username}?start={id}"
153
154 await message.answer(_("Выберите категорию продуктов:"),
155 reply_markup=menu_markup)
156
157 # Показываем список доступных товаров
158 @dp.callback_query_handler(text_contains="sushi")
159 async def show_items(call: CallbackQuery):
160
161     all_items = await db.show_items()
162     # Проходимся по товарам, пронумеровывая
163     # for item in all_items:
164     text = _("<b>Товар</b>: <i>{name}</i>\n"
165             "<b>Описание</b>: <i>{description}</i>\n"
166             "<b>Цена:</b> \t{price:,} грн.\n")
167     for num, item in enumerate(all_items):
168
169         markup = InlineKeyboardMarkup(
170             inline_keyboard=
171             [
172                 [
173                     # Создаем кнопку "купить" и передаем ее айдишник в функцию создания коллбека
174                     InlineKeyboardButton(text=_("Добавить в корзину"),
175 callback_data=buy_item.new(item_id=item.id)),
176                     # InlineKeyboardButton(text="Добавить в корзину",
177 callback_data="to_korsina")
178                 ],
179             ]
180         )
181
182     # Отправляем фотку товара с подписью и кнопкой "купить"
183     await call.message.answer_photo(
184         photo=item.photo,
185         caption=text.format(

```

```

186         id=item.id,
187         name=item.name,
188         description=item.description,
189         price=item.price / 100
190     ),
191     reply_markup=markup
192 )
193 # Между сообщениями делаем небольшую задержку, чтобы не упереться в
194 лимиты
195 await asyncio.sleep(0.3)
196
197 # Для фильтрации по коллбекам можно использовать buy_item.filter()
198 @dp.callback_query_handler(buy_item.filter())
199 async def buying_item(call: CallbackQuery, callback_data: dict, state:
200 FSMContext):
201     # То, что мы указали в CallbackData попадает в хендлер под callback_data,
202 как словарь, поэтому достаем айдишник
203     item_id = int(callback_data.get("item_id"))
204     await call.message.edit_reply_markup()
205
206     # Достаем информацию о товаре из базы данных
207     item = await database.Item.get(item_id)
208     if not item:
209         await call.message.answer(_("Такого товара не существует"))
210     return
211
212     text = _("Вы хотите заказать товар \<b>{name}</b>\<b>\" по цене:
213 <b>{price:,}</b>/шт.</b>\n\"
214     "Введите количество или нажмите отмена").format(name=item.name,
215 price=item.price / 100)
216     await call.message.answer(text)
217     await states.Purchase.EnterQuantity.set()
218
219     # Сохраняем в ФСМ класс товара и покупки
220     await state.update_data(
221         item=item,
222         purchase=database.Purchase(
223             item_id=item_id,
224             purchase_time=datetime.datetime.now(),
225             buyer=call.from_user.id
226         )
227     )
228
229 # Принимаем в этот хендлер только цифры
230 @dp.message_handler(regex=r"^\d+$", state=states.Purchase.EnterQuantity)
231 async def enter_quantity(message: Message, state: FSMContext):
232     # Получаем количество указанного товара
233     quantity = int(message.text)
234     async with state.proxy() as data: # Работаем с данными из ФСМ
235         data["purchase"].quantity = quantity
236         item = data["item"]
237         amount = item.price * quantity
238         data["purchase"].amount = amount
239
240     # Создаем кнопки
241     agree_button = InlineKeyboardButton(
242         text=_("Ок"),
243         callback_data="agree"
244     )
245     change_button = InlineKeyboardButton(
246         text=_("Заново"),
247         callback_data="change"
248     )
249     cancel_button = InlineKeyboardButton(

```

```

250     text=_("Отмена"),
251     callback_data="cancel"
252 )
253
254 # Создаем клавиатуру
255 markup = InlineKeyboardMarkup(
256     inline_keyboard=
257     [
258         [agree_button, change_button, cancel_button] # ряд кнопок
259     ]
260 )
261 await message.answer(
262     _("Хорошо, вы хотите купить <i>{quantity}</i> {name} по цене
263 <b>{price:,}</b>/шт.</b>\n\n"
264     "Получится <b>{amount:,}</b>. Подтверждаете?").format(
265         quantity=quantity,
266         name=item.name,
267         amount=amount / 100,
268         price=item.price / 100
269     ),
270     reply_markup=markup)
271 await states.Purchase.Approval.set()
272
273 # То, что не является числом - не попало в предыдущий хендлер и попадает в этот
274 @dp.message_handler(state=states.Purchase.EnterQuantity)
275 async def not_quantity(message: Message):
276     await message.answer(_("Неверное значение, введите число"))
277
278 # Если человек нажал на кнопку Отменить во время покупки - убираем все
279 @dp.callback_query_handler(text_contains="cancel", state=states.Purchase)
280 async def approval(call: CallbackQuery, state: FSMContext):
281     await call.message.edit_reply_markup() # Убираем кнопки
282     await call.message.answer(_("Вы отменили эту покупку"))
283     await state.reset_state()
284
285 # Если человек нажал "ввести заново"
286 @dp.callback_query_handler(text_contains="change",
287 state=states.Purchase.Approval)
288 async def approval(call: CallbackQuery):
289     await call.message.edit_reply_markup() # Убираем кнопки
290     await call.message.answer(_("Введите количество товара заново. "))
291     await states.Purchase.EnterQuantity.set()
292
293 # Если человек нажал "согласен"
294 @dp.callback_query_handler(text_contains="agree",
295 state=states.Purchase.Approval)
296 async def approval(call: CallbackQuery, state: FSMContext):
297     await call.message.edit_reply_markup() # Убираем кнопки
298
299     data = await state.get_data()
300     purchase = data.get("purchase")
301     item = data.get("item")
302     # Теперь можно внести данные о покупке в базу данных через .create()
303     await purchase.create()
304
305     markup = InlineKeyboardMarkup(
306         inline_keyboard=
307         [
308             [
309                 InlineKeyboardButton(text=_("Оформить покупку"),
310 callback_data="buy"),
311                 InlineKeyboardButton(text=_("Корзина"), callback_data="cart")
312             ],
313         ]

```



```

314     )
315
316     await call.message.answer(_("Хорошо, товар добавлен в корзину. К оплате
317 <b>{amount:,}</b> грн.
318 ).format(amount=purchase.amount / 100), reply_markup=markup)
319     await states.Purchase.EnterAdress.set()
320
321     # Для фильтрации по коллбекам можно использовать buy_item.filter()
322 @dp.callback_query_handler(text_contains="buy", state=states.Purchase.EnterAdress
323 )
324     async def address(call: CallbackQuery, state: FSMContext):
325         await call.message.answer(_("Хорошо, ваш адресс?"))
326         await states.Purchase.EnterAdress1.set()
327
328 @dp.message_handler(state=states.Purchase.EnterAdress1)
329     async def enter_quantity(message: Message, state: FSMContext):
330         shipping_address = message.text
331         async with state.proxy() as data: # Работаем с данными из ФСМ
332             data["purchase"].shipping_address = shipping_address
333
334         agree_button = InlineKeyboardButton(
335             text=_("ОК"),
336             callback_data="agree1")
337         change_button = InlineKeyboardButton(
338             text=_("Заново"),
339             callback_data="change")
340         cancel_button = InlineKeyboardButton(
341             text=_("Отмена"),
342             callback_data="cancel")
343
344         # Создаем клавиатуру
345         markup = InlineKeyboardMarkup(
346             inline_keyboard=
347             [
348                 [agree_button, change_button, cancel_button] # Первый
349             ])
350         await call.message.answer(
351             _("Хорошо, ваш адресс <i>{shipping_address}</i>\n\n"
352             "Подтверждаете?").format(
353                 shipping_address=shipping_address), reply_markup=markup)
354         await states.Purchase.Approval1.set()
355
356     # Если человек нажал "согласен"
357 @dp.callback_query_handler(text_contains="agree1",
358 state=states.Purchase.Approval1)
359     async def get_phone(call: CallbackQuery, state: FSMContext):
360         await call.message.edit_reply_markup() # Убираем кнопки
361
362
363         data = await state.get_data()
364         purchase = data.get("purchase")
365         item = data.get("item")
366         # Теперь можно внести данные о покупке в базу данных через .create()
367         await purchase.create()
368
369         markup = types.ReplyKeyboardMarkup(
370             keyboard=
371             [
372                 [
373                     types.KeyboardButton(text="Поделиться телефоном", request_contact=True)
374                 ]
375             ],
376             resize_keyboard=True,
377             one_time_keyboard=True

```

```

378         )
379
380         await call.message.answer(
381             _("Хорошо, введите ваш телефон или нажмите клавишу
382 поделиться номером?"), reply_markup=markup)
383         await states.Purchase.EnterPhone.set()
384
385     @dp.message_handler(state=states.Purchase.EnterPhone)
386     async def enter_quantity(message: Message, state: FSMContext):
387
388         phone_number = message.text
389         async with state.proxy() as data: # Работаем с данными из
390     ФСМ
391             data["purchase"].phone_number = phone_number
392
393             # Создаем кнопки
394             agree_button = InlineKeyboardButton(
395                 text=_("Согласен"),
396                 callback_data="agree2")
397
398             cancel_button = InlineKeyboardButton(
399                 text=_("Отменить покупку"),
400                 callback_data="cancel")
401
402             # Создаем клавиатуру
403             markup = InlineKeyboardMarkup(
404                 inline_keyboard=
405                 [
406                     [agree_button, cancel_button] # Третий ряд
407                 ])
408             await call.message.answer(
409                 _("Хорошо, ваш телефон <i>{phone_number}</i>\n\n"
410                 "Подтверждаете покупку?").format(
411                     phone_number=phone_number
412                 ), reply_markup=markup)
413             await states.Purchase.Approval2.set()
414
415     @dp.callback_query_handler(text_contains="agree2", state=states.Purchase.Approval
416     2)
417         async def get_phone(call: CallbackQuery, state: FSMContext):
418             await call.message.edit_reply_markup()
419             data = await state.get_data()
420             purchase = data.get("purchase")
421             item = data.get("item")
422             # Теперь можно внести данные о покупке в базу данных через .create()
423             await purchase.create()
424             await call.message.answer(
425                 _("Хорошо, введите ваш заказ оформлен! Спасибо
426 за покупку!!!"))
427
428     @dp.message_handler()
429     async def echo(m: Message):
430         await m.answer(m.text)
431
432     async def check_payment(purchase: database.Purchase):
433         return True
434
435

```

## Лістинг коду admin\_panel.py

```

1 from aiogram import types
2 from aiogram.dispatcher import FSMContext
3 from aiogram.types import InlineKeyboardMarkup, InlineKeyboardButton
4 from config import admin_id
5 from load_all import dp, _, bot
6 from states importNewItem, Mailing
7 from database import Item, User
8 @dp.message_handler(user_id=admin_id, commands=["cancel"], state=NewItem)
9 async def cancel(message: types.Message, state: FSMContext):
10     await message.answer(_("Вы отменили создание товара"))
11     await state.reset_state()
12 @dp.message_handler(user_id=admin_id, commands=["add_item"])
13 async def add_item(message: types.Message):
14     await message.answer(_("Введите название товара или нажмите /cancel"))
15     awaitNewItem.Name.set()
16 @dp.message_handler(user_id=admin_id, state=NewItem.Name)
17 async def enter_name(message: types.Message, state: FSMContext):
18     name = message.text
19     item = Item()
20     item.name = name
21     await message.answer(_("Название: {name}"
22                             "\nПришлите мне описание товара или нажмите
23 /cancel").format(name=name
24 ))
25     awaitNewItem.Description.set()
26     await state.update_data(item=item)
27 @dp.message_handler(user_id=admin_id, state=NewItem.Description)
28 async def enter_Description(message: types.Message, state: FSMContext):
29     description = message.text
30     data = await state.get_data()
31     item: Item = data.get("item")
32     item.description = description
33     await message.answer(_("Название: {name}"
34                             "\nОписание: {description}"
35                             "\nПришлите мне фотографию товара (не документ) или
36 нажмите/cancel").format(
37         name=item.name,
38         description=description
39     ))
40     awaitNewItem.Photo.set()
41     await state.update_data(item=item)
42 @dp.message_handler(user_id=admin_id, state=NewItem.Photo,
43 content_types=types.ContentType.PHOTO)
44 async def add_photo(message: types.Message, state: FSMContext):
45     photo = message.photo[-1].file_id
46     data = await state.get_data()
47     item: Item = data.get("item")
48     item.photo = photo
49     await message.answer_photo(
50         photo=photo,
51         caption=_("Название: {name}"
52                 "\nНазвание: {description}"
53                 "\nПришлите мне цену товара в копейках или нажмите
54 /cancel").format(name=item.name,
55                 description=item.description
56 ))
57     awaitNewItem.Price.set()
58     await state.update_data(item=item)
59 @dp.message_handler(user_id=admin_id, state=NewItem.Price)
60 async def enter_price(message: types.Message, state: FSMContext):

```

```

61     data = await state.get_data()
62     item: Item = data.get("item")
63     try:
64         price = int(message.text)
65     except ValueError:
66         await message.answer(_("Неверное значение, введите число"))
67         return
68     item.price = price
69     markup = types.InlineKeyboardMarkup(
70         inline_keyboard=
71         [
72             [InlineKeyboardButton(text=_("Да"), callback_data="confirm")],
73             [InlineKeyboardButton(text=_("Ввести заново"),
74 callback_data="change")],
75         ]
76     )
77     await message.answer(_("Цена: {price:,}\n"
78                             "Подтверждаете? Нажмите /cancel чтобы
79 отменить").format(price=price / 100), reply_markup=markup)
80     await state.update_data(item=item)
81     await NewItem.Confirm.set()
82 @dp.callback_query_handler(user_id=admin_id, text_contains="change",
83 state=NewItem.Confirm)
84 async def change_price(call: types.CallbackQuery):
85     await call.message.edit_reply_markup()
86     await call.message.answer(_("Введите заново цену товара в копейках"))
87     await NewItem.Price.set()
88 @dp.callback_query_handler(user_id=admin_id, text_contains="confirm",
89 state=NewItem.Confirm)
90 async def confirm(call: types.CallbackQuery, state: FSMContext):
91     await call.message.edit_reply_markup()
92     data = await state.get_data()
93     item: Item = data.get("item")
94     await item.create()
95     await call.message.answer(_("Товар удачно создан.))
96     await state.reset_state()
97 # Фича для рассылки по юзерам (учитывая их язык)
98 @dp.message_handler(user_id=admin_id, commands=["tell_everyone"])
99 async def mailing(message: types.Message):
100     await message.answer(_("Пришлите текст рассылки"))
101     await Mailing.Text.set()
102 @dp.message_handler(user_id=admin_id, state=Mailing.Text)
103 async def enter_text(message: types.Message, state: FSMContext):
104     text = message.text
105     await state.update_data(text=text)
106     markup = InlineKeyboardMarkup(
107         inline_keyboard=
108         [
109             [InlineKeyboardButton(text="Русский", callback_data="ru")],
110             [InlineKeyboardButton(text="English", callback_data="en")],
111             [InlineKeyboardButton(text="Українська", callback_data="uk")],
112         ]
113     )
114     await message.answer(_("Пользователям на каком языке разослать это
115 сообщение?\n\n"
116                             "Текст:\n"
117                             "{text}").format(text=text),
118                             reply_markup=markup)
119     await Mailing.Language.set()
120 @dp.callback_query_handler(user_id=admin_id, state=Mailing.Language)
121 async def mailing_start(call: types.CallbackQuery, state: FSMContext):
122     data = await state.get_data()
123     text = data.get("text")
124     await state.reset_state()

```

```

125     await call.message.edit_reply_markup()
126     users = await User.query.where(User.language == call.data).gino.all()
127     for user in users:
128         try:
129             await bot.send_message(chat_id=user.user_id, text=text)
130             await sleep(0.3)
131         except Exception:
132             pass
133     await call.message.answer(_("Рассылка выполнена."))

```

### Лістинг коду app.py

```

1  import asyncio
2  from aiogram import executor
3  from config import admin_id
4  from database import create_db
5  from load_all import bot
6  async def on_shutdown(dp):
7  await bot.close()
8  async def on_startup(dp):
9  await create_db()
10 await bot.send_message(admin_id, "Я запущен!")
11 if __name__ == '__main__':
12 from admin_panel import dp
13 from handlers import dp16
14 executor.start_polling(dp, on_shutdown=on_shutdown, on_startup=on_startup)

```

### Лістинг коду load\_all.py

```

1
2  import asyncio
3  import logging
4  from aiogram import Bot
5  from aiogram import Dispatcher
6  from aiogram.contrib.fsm_storage.memory import MemoryStorage
7  from config import TOKEN
8  from language_middleware import setup_middleware
9
10 logging.basicConfig(format=u'%(filename)s [LINE:%(lineno)d] #%(levelname)-8s
11 [% (asctime)s] % (message)s',
12                     level=logging.INFO)
13 storage = MemoryStorage()
14
15 bot = Bot(token=TOKEN, parse_mode="HTML")
16 dp = Dispatcher(bot, storage=storage)

```

## Додаток Б Презентація

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ

### Програмний модуль для автоматизації надання послуг в закладах громадського харчування

Виконала:  
Студентка групи КІ-16д  
Гусейнова Олена Сергіївна

Керівник проекту:  
к.т.н., доцент  
Деркач Марина Володимирівна

Рисунок Б.1 – Слайд №1

## Актуальність

- Стрімкий ріст кількості закладів громадського харчування
- Підвищення попиту на користування онлайн-сервісами
- Автоматизація надання послуги доставки
- Отримання послуг зручним та простим способом



Рисунок Б.2 – Слайд №2

## Аналіз існуючих сервісів



Рисунок Б.3 – Слайд №3

## Постановка завдання



**Мета:** розробка програмного модулю для розширення функціональних можливостей надання послуг у закладі громадського харчування



**Об'єкт дослідження** – методи й засоби автоматизації надання послуг клієнтам в закладах громадського харчування

Рисунок Б.4 – Слайд №4

## Вибір засобів розробки



PostgreSQL



Рисунок Б.5 – Слайд №5

## Структура програмного модулю

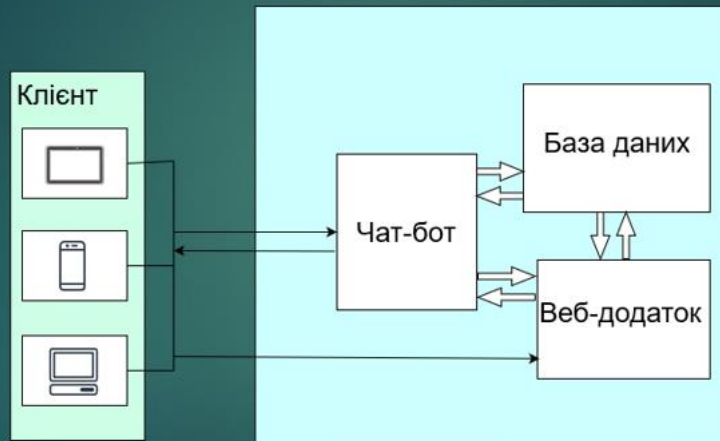


Рисунок Б.6 – Слайд №6



## Декомпозиція створення програмного модулю

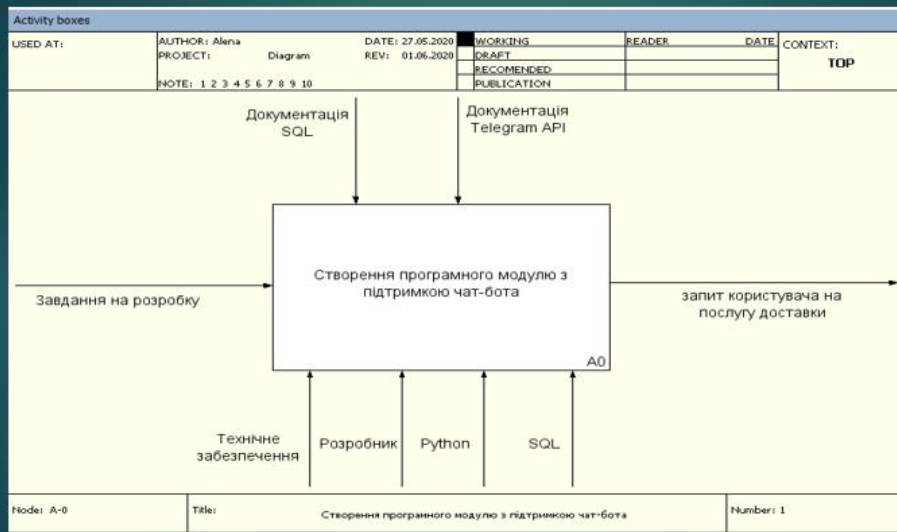


Рисунок Б.7 – Слайд №7

## Діаграма

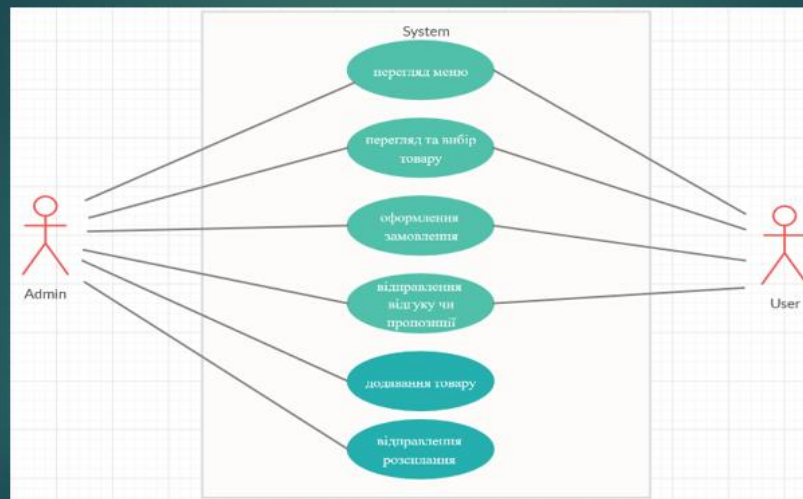


Рисунок Б.8 – Слайд №8



Рисунок Б.9 – Слайд №9

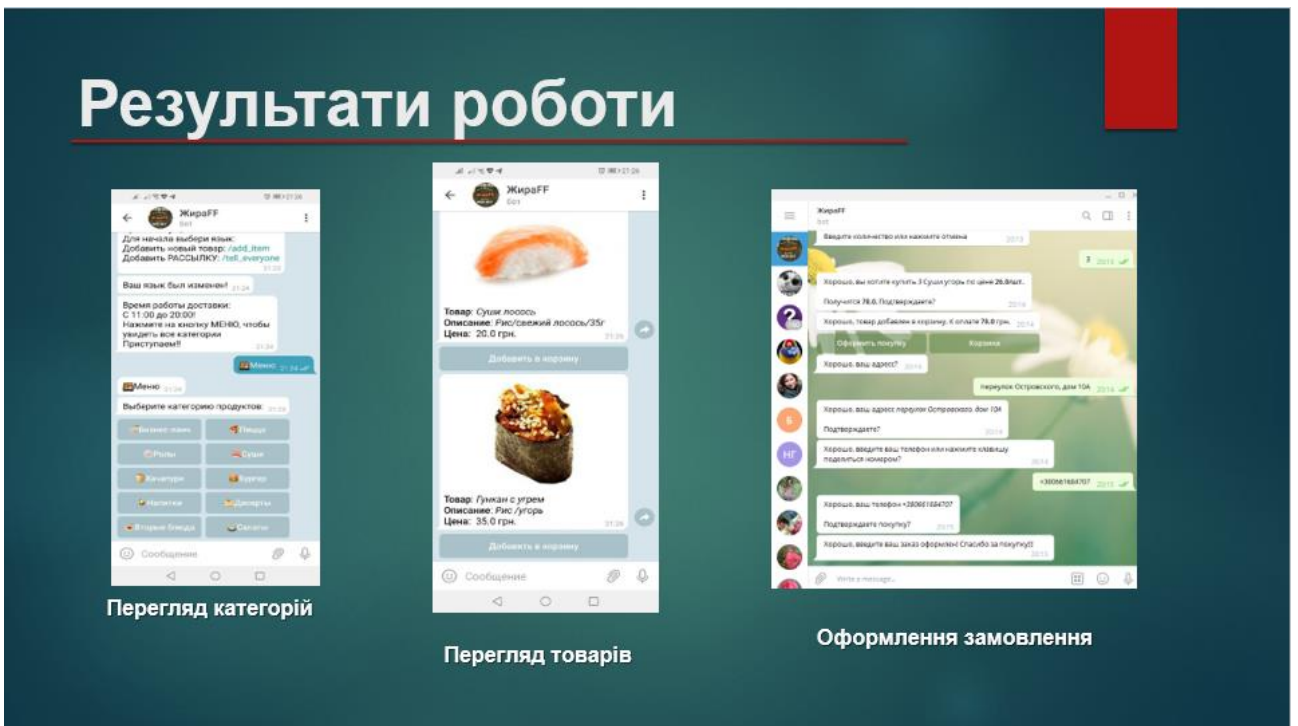


Рисунок Б.10 – Слайд №10

## Результати роботи

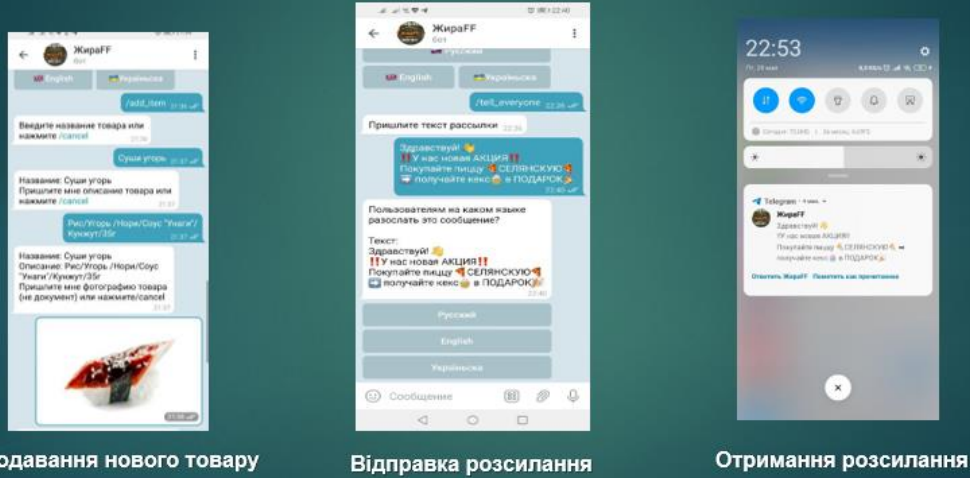


Рисунок Б.11 – Слайд №11

## Висновки

- ✓ Проведений аналіз актуальності теми розробки та існуючих сервісів
- ✓ Розроблений програмний модуль у вигляді підтримки чат-боту Телеграм
- ✓ Автоматизуються такі процеси:
  - формування заявки користувача на замовлення доставки продуктів
  - автоматичне повідомлення менеджера про заявку
  - розсилання клієнтам повідомлення про акції та нові пропозиції

Рисунок Б.11 – Слайд №12