

# **ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЄКТ**

**на тему:**

**«Проектування програмного забезпечення системи  
стратегічного аналізу діяльності call-центру»**

Студентки 2 курсу, 23 групи,  
спеціальності 121 «Інженерія  
програмного забезпечення»  
спеціалізації «Інженерія  
програмного забезпечення»

\_\_\_\_\_

підпис студента

**Бойко  
Анни Анатоліївни**

Науковий керівник  
кандидат технічних наук,  
доцент кафедри інженерії  
програмного забезпечення та  
кібербезпеки

\_\_\_\_\_

підпис керівника

**Рзаєва Світлана  
Леонідівна**

Гарант освітньої програми  
доктор економічних наук,  
професор кафедри інженерії  
програмного забезпечення та  
кібербезпеки

\_\_\_\_\_

підпис гаранта

**Токар Володимир  
Володимирович**

# Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Спеціальність 121 «Інженерія програмного забезпечення»

**Затверджую**

Зав. кафедри інженерії програмного  
забезпечення та кібербезпеки

Криворучко О. В.

«10» листопада 2020 р.

## **Завдання**

### **на випускний кваліфікаційний проєкт студентів**

Бойко Анни Анатоліївни

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «Проектування програмного  
забезпечення системи стратегічного аналізу діяльності call-центру»

Затверджена наказом ректора від «28» грудня 2020 р. № 3923

2. Строк здачі студентом закінченого проєкту 25 листопада 2021

3. Цільова установка та вихідні дані до проєкту  
Мета проєкту є проектування стратегічного аналізу діяльності call-центру  
з використанням веб-додатку

Об'єкт дослідження діяльність контакт-центру

Предмет дослідження розробка проектування і реалізація системи стратегічного  
аналізу діяльності call-центру

4. Консультанти проєкту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проєкту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Поняття call-центр, сутність і основні характеристики

1.2. Сутність і основні характеристики поняття інформаційної системи

1.3. Сутність поняття веб-сайту

1.4. Процес розробки веб-сайтів, мета і методології

1.5. Висновок до розділу 1

РОЗДІЛ 2. ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ, ОПИС ТА ОБГРУНТУВАННЯ ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1. Постановка задачі розробки

2.2. Вибір мови програмування

2.3. Вибір середовища розробки інформаційної системи

2.4. Вибір системи керування базами даних

2.5. Вибір фреймворку веб-розробки для мови Python

2.6. Висновок до розділу 2

РОЗДІЛ 3. ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОЗРОБЛЮВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1. Структурна діаграма

3.2. Діаграма варіантів використання інформаційної системи

3.3. Висновок до розділу 3

РОЗДІЛ 4. РЕАЛІЗАЦІЯ СИСТЕМИ СТРАТЕГІЧНОГО АНАЛІЗУ ДІЯЛЬНОСТІ CALL-ЦЕНТРУ

4.1. Діаграма класів

4.2. Розробка графічного інтерфейсу користувача

4.3. Аналіз контрольного прикладу

4.4. Висновок до розділу 4

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

## 6. Календарний план виконання проєкту

№ пор.	Назва етапів випускного кваліфікаційного проєкту	Строк виконання етапів проєкту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проєкту</i>	21.09.2020	21.09.2020
2.	<i>Розробка та затвердження завдання на проєкт магістра</i>	29.12.2020	22.12.2020
3.	<i>Вступ та перелік літературних джерел</i>	27.02.2021	27.02.2021
4.	<i>Розробка технічного завдання</i>	20.03.2021	20.03.2021
5.	<i>Розділ 1. Аналіз предметної області</i>	16.04.2021	16.04.2021
6.	<i>Розділ 2. Постановка задачі розробки, опис та обґрунтування засобів реалізації</i>	24.05.2021	24.05.2021
7.	<i>Розділ 3. Загальна характеристика розроблюваної інформаційної системи</i>	21.06.2021	21.06.2021
8.	<i>Розділ 4. Реалізація системи стратегічного аналізу діяльності call-центру</i>	20.09.2021	20.09.2021
9.	<i>Розробка програми та методики тестування</i>	18.10.2021	18.10.2021
10.	<i>Написання наукової статті</i>	22.05.2021	22.05.2021
11.	<i>Керівництво користувача</i>	21.10.2021	21.10.2021
12.	<i>Висновки та пропозиції</i>	01.11.2021	01.11.2021
13.	<i>Здача випускного кваліфікаційного проєкту на кафедрі (перша перевірка)</i>	03.11.2021	03.11.2021
14.	<i>Підготовка автореферату та презентації доповіді</i>	03.11.2021	03.11.2021
15.	<i>Попередній захист випускного кваліфікаційного проєкту</i>	22.11.2021 – 25.11.2021	22.11.2021
16.	<i>Здача зброшурованої випускного кваліфікаційного проєкту</i>	25.11.2021	25.11.2021
17.	<i>Зовнішнє рецензування випускного кваліфікаційного проєкту</i>	26.11.2021	26.11.2021
18.	<i>Підготовка до публічного захисту випускного кваліфікаційного проєкту</i>		

7. Дата видачі завдання «29» грудня 2020 р.

8. Науковий керівник випускного кваліфікаційного проєкту Рзаєва С.Л.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми Токар В.В.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Бойко А.А.

(прізвище, ініціали, підпис)

## 11. Відгук керівника випускного кваліфікаційного проєкту

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Науковий керівник випускного кваліфікаційного проєкту \_\_\_\_\_

*(підпис, дата)*

Відмітка про попередній захист \_\_\_\_\_

*(ПІБ, підпис, дата)*

## 12. Висновок про випускний кваліфікаційний проєкт

Випускний кваліфікаційний проєкт студента \_\_\_\_\_

Бойко А.А.

*(прізвище, ініціали)*

може бути допущена до захисту екзаменаційній комісії.

Гарант освітньої програми \_\_\_\_\_

Токар В.В.

*(прізвище, ініціали, підпис)*

Завідувач кафедри \_\_\_\_\_

Криворучко О. В.

*(підпис, прізвище, ініціали)*

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_\_\_ р.

## АНОТАЦІЯ

Відповідно до мети дослідження робота присвячена моделі розроблення веб-додатку для стратегічного аналізу діяльності call-center. В процесі роботи створено інформаційну систему стратегічного аналізу діяльності call-центру. Дана система покликана виконувати функціонал електронного контролю успішності роботи операторів і їх розвитку.

Розробка інформаційного стратегічного аналізу виконана в PyCharm.

PyCharm – це інтегроване середовище розробки (IDE), що використовується в комп'ютерному програмуванні, зокрема для мови Python.

**Ключові слова:** *веб-додаток, контакт-центр, стратегічний аналіз, хмарні обчислення, аутсорсинг, інформаційна система, Python, геоінформаційні системи, PyCharm.*

## ABSTRACT

In accordance with the purpose of the study, the work is devoted to the model of web application development for strategic analysis of the call center.

In the process of the work the information system of strategic analysis of the contact center activity was created. The system is intended to perform the function of electronic control of the operators' work success and their development.

The development of informational strategic analysis is done in PyCharm. PyCharm. is an integrated development environment (IDE), which is used in computer programming, especially for Python language.

**Keywords:** *web application, contact center, strategic analysis, cloud computing, outsourcing, information system, Python, geoinformation systems, PyCharm.*

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення

ІС – інформаційна система

XML (англ. Extensible Markup Language) – розширювана мова розмітки

АТС – автоматична телефонна станція

PSTN (англ. Public Switched Telephone Network) – телефонна мережа загального користування

IP (англ. Internet Protocol) – міжмережевий протокол

ACD (англ. Automatic Call Distribution) – система автоматичного розподілу вхідних та вихідних звернень

API (англ. Application Programming Interface) - набір функцій, який дозволяє додаткам отримувати доступ до даних і взаємодіяти із зовнішніми програмними компонентами

SaaS (англ. Software as a Service) – програмне забезпечення як послуга

CRM (англ. Customer Relationship Management) – система управління взаємовідносинами з клієнтами

CTI (англ. Computer Telephony Integration) інтеграція комп'ютерної телефонії

GUI (англ. Graphical user interface) – графічний інтерфейс користувача

RSPCA (англ. Royal Society for the Prevention of Cruelty to Animals)

CMS (англ. Content Management System) – система керування вмістом

SQL (англ. Structured Query Language) – мова структурованих запитів

ASP (англ. Active Server Pages) – активні серверні сторінки

JSP (англ. Active Server Pages) – це не стандартні html-сторінки

PHP (англ. Hypertext Preprocessor) скриптова мова програмування

<i>КНТЕУ 121 02 з - 03. МР</i>				
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>
Зав. каф.		Криворучко О.В.		29.09.20
Керівник		Рзаєва С.Л.		29.09.20
Гарант		Токар В.В.		29.09.20
Розробив		Бойко А.А.		29.09.20
<i>Проектування ПЗ системи стратегічного аналізу діяльності call-центру</i>				
<i>Перелік умовних скорочень</i>				
<i>Стадія</i>		<i>Аркуш</i>		<i>Аркуші</i>
<i>ПС</i>		<i>2</i>		<i>48</i>
<i>Факультет інформаційних технологій 2 курс, 23 група</i>				

## ЗМІСТ

<b>ВСТУП .....</b>	<b>3</b>
<b>РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....</b>	<b>6</b>
1.1. Поняття call-центр, сутність і основні характеристики .....	6
1.2. Сутність і основні характеристики поняття інформаційної системи.....	10
1.3. Сутність поняття веб-сайту.....	15
1.4. Процес розробки веб-сайтів, мета і методології .....	16
1.5. Висновок до розділу 1.....	19
<b>РОЗДІЛ 2 ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ, ОПИС ТА ОБГРУНТУВАННЯ ЗАСОБІВ РЕАЛІЗАЦІЇ .....</b>	<b>20</b>
2.1. Постановка задачі розробки.....	20
2.2. Вибір мови програмування.....	20
2.3. Вибір середовища розробки інформаційної системи.....	26
2.4. Вибір системи керування базами даних.....	26
2.5. Вибір фреймворку веб-розробки для мови Python.....	29
2.6. Висновок до розділу 2.....	29
<b>РОЗДІЛ 3 ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОЗРОБЛЮВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....</b>	<b>30</b>
3.1. Структурна діаграма .....	30
3.2. Діаграма варіантів використання інформаційної системи.....	31
3.3. Висновок до розділу 3.....	34
<b>РОЗДІЛ 4 РЕАЛІЗАЦІЯ СИСТЕМИ СТРАТЕГІЧНОГО АНАЛІЗУ ДІЯЛЬНОСТІ CALL- ЦЕНТРУ</b>	
4.1. Діаграма класів .....	35
4.2. Розробка графічного інтерфейсу користувача.....	36
4.3. Аналіз контрольного прикладу .....	41
4.4. Висновок до розділу 4.....	45
<b>ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....</b>	<b>46</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>47</b>
<b>ДОДАТКИ.....</b>	<b>49</b>

					<i>КНТЕУ 121 02з-03.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата	Проектування ПЗ системи стратегічного аналізу діяльності call-центру	Стадія	Аркуш	Аркуші
Зав. каф.		Криворучко О.В.		22.12.20		3	3	48
Керівник		Рзаєва С.Л.		22.12.20		Факультет інформаційних технологій		
Гарант		Токар В.В.		22.12.20		2 курс, 2з група		
Розробив		Бойко А.А.		22.12.20		Зміст		



## ВСТУП

*Актуальність.* Кожного дня в сучасному світі все активніше розвивається тенденція діджиталізації і автоматизації різного роду процесів. Сучасники намагаються перевести в електронний формат всі можливі взаємодії і сховища інформації.

Майже у кожного закладу чи підприємства є в наявності своя інформаційна система, яка зачасту розрахована не тільки на взаємодію зі співробітниками, але й користування з боку клієнтів.

Подібні тенденції не обійшли й одну з найважливіших сфер людського життя, а саме сферу обслуговування клієнтів. Все більше різноманітних закладів проектують власні інформаційні системи і портали, для внутрішньої взаємодії між співробітниками.

Мета дослідження розроблення веб-додатку стратегічного аналізу діяльності call-центру.

Об'єкт дослідження: інформаційні системи підприємства.

Предмет дослідження проектування інформаційної системи стратегічного аналізу діяльності call-center.

Для досягнення поставленої мети слід виконати наступні завдання:

- проаналізувати діяльність call-center;
- проаналізувати методи розроблення веб-сайтів;
- обрати засоби реалізації моделі стратегічного аналізу;
- дати загальну характеристику розроблюваної інформаційної системи;
- побудувати діаграму класів;

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>КНТЕУ 121 023-03.МР</i>			
Зав. каф.		Криворучко О.В.		27.02.21	<i>Проектування ПЗ системи стратегічного аналізу діяльності call-центру</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Рзасва С.Л.		27.02.21		<i>В</i>	<i>4</i>	<i>48</i>
Гарант		Токар В.В.		27.02.21		<i>Факультет інформаційних технологій 2 курс, 23 група</i>		
Розробив		Бойко А.А.		27.02.21				
					<i>Вступ</i>			

- розробити графічний інтерфейс користувача;
- провести аналіз контрольного прикладу.

*Методи дослідження* — метод комп'ютерного моделювання і проектування.

З *практичної точки зору*, розроблену в ході роботи систему можна впроваджувати на базі call-center, отримуючи повноцінну інформаційну систему, яка здатна виконувати функціонал електронного контролю успішності роботи операторів і їх розвитку.

Впровадження не несе за собою великих витрат та не потребує особливих навичок умінь. Основна задача працівників установи у випадку впровадження даної системи обліку успішності — перенести дані з паперових носіїв в систему і робити це регулярно.

					<i>КНТЕУ 121 023-03.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		5



Дедалі частіше передача голосу та даних до центру пов'язується за допомогою набору нових технологій, що називається інтеграцією комп'ютерної телефонії.

Call-center - це центральний пункт, з якого здійснюється управління всіма контактами клієнтів. Через контактні центри цінна інформація про компанію передається відповідним людям, контакти відстежуються та збираються дані. Як правило, це частина інфраструктури управління відносинами з клієнтами компанії. Більшість великих компаній використовують контактні центри як засіб управління взаємодією з клієнтами. Цими центрами може керувати як внутрішній відділ, що відповідає, так і передача клієнтам сторонніх агентств стороннім агентствам (відомим як Аутсорсингові телефонні центри).

Технології call-center включають: програмне забезпечення розпізнавання мовлення, що дозволило системам інтерактивної голосової відповіді (IVR) обробляти перші рівні підтримки клієнтів, видобуток тексту, обробку природних мов, щоб забезпечити кращу обробку клієнтів, навчання агентів за допомогою інтерактивних сценаріїв та автоматичний майнінг з використанням найкращих практик минулого взаємодії, автоматизація підтримки та багато інших технологій для підвищення продуктивності агента та задоволеності споживачів. Автоматичний вибір свинцю або керування свинцем також призначений для підвищення ефективності як для вхідних, так і для вихідних кампаній. Це дозволяє безпосередньо направляти вхідні дзвінки до відповідного агента для завдання, мінімізуючи час очікування та довгі списки нерелевантних опцій для людей, що телефонують.

Що стосується вихідних дзвінків, вибір потенційних клієнтів дозволяє керівництву визначити, який тип потенційних клієнтів надходить до того чи

						Аркуш
						7
Зм.	Аркуш	№ докум	Підпис	Дата	<i>КНТЕУ 121 023-03.МР</i>	

іншого агента на основі таких факторів, як кваліфікація, соціально-економічні фактори, минулі результати та відсоткова ймовірність закриття продажу на потенційного клієнта.

Універсальна черга стандартизує обробку комунікацій за різними технологіями, такими як факс, телефон та електронна пошта. Віртуальна черга надає абонентам альтернативу очікуванню на утриманні, коли відсутні агенти для обробки запиту на вхідний дзвінок.

до відповідного агента для завдання, мінімізуючи час очікування та довгі списки нерелевантних опцій для людей, що телефонують.

Історично, call-центри будувались на обладнанні приватних філій (АТС), яке належить, розміщується та обслуговується оператором call-центру. АТС може надавати такі функції, як автоматичний розподіл дзвінків, інтерактивна голосова відповідь та маршрутизація на основі навичок.

У моделі віртуального call-центру оператор call-центру (бізнес) сплачує щомісячну або річну плату постачальнику, який розміщує телефонну службу call-центру та обладнання передачі даних у власному об'єкті на базі хмарних технологій. У цій моделі оператор не володіє, не експлуатує та не приймає обладнання, на якому працює call-центр. Агенти підключаються до обладнання постачальника через традиційні телефонні лінії PSTN або через голос через IP. Дзвінки до або з потенційних клієнтів або контактів надходять або закінчуються в центрі обробки даних постачальника, а не в приміщенні оператора call-центру. Потім обладнання постачальника телефонії (часом сервери даних) підключає дзвінки до агентів оператора call-центру. [16]

Технологія віртуального call-центру дозволяє людям працювати вдома або в будь-якому іншому місці, а не в традиційному, централізованому центрі call-центру, що все більше дозволяє людям "в дорозі" або з фізичними чи іншими вадами працювати з потрібних місць - тобто не виїжджати їх будинок.

					<i>КНТЕУ 121 023-03.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

Єдиним необхідним обладнанням є доступ до Інтернету та робоча станція. [17] Компанії віддають перевагу послугам віртуального телефонного центру через перевагу у вартості. Компанії можуть негайно розпочати свій бізнес-центр, не встановлюючи базової інфраструктури, як Dialer, ACD та IVRS. [18]

Віртуальні call-центри стали все частіше використовуватися після того, як пандемія COVID-19 обмежила бізнес у роботі з великими групами людей, які працюють в безпосередній близькості.

#### Хмарні обчислення

Завдяки використанню інтерфейсів прикладного програмування (API), розміщені та вимогові call-центри, побудовані на хмарному програмному забезпеченні як послугі (SaaS), можуть інтегрувати свою функціональність із хмарними додатками для управління взаємовідносинами з клієнтами (CRM), управління свинцем та інше.

Розробники використовують API для вдосконалення функціональних можливостей платформи центру викликів на базі хмарних технологій, включаючи API інтеграції комп'ютерної телефонії (CTI), які забезпечують базові засоби управління телефонією та складну обробку викликів з окремого додатка, та API конфігурації, що дозволяють керувати графічними інтерфейсами користувача (GUI) адміністративних функцій.

#### Аутсорсинг

Аутсорсингові call-центри часто розташовані в країнах, що розвиваються, де заробітна плата значно нижча. Сюди входять галузі call-центрів на Філіппінах, Бангладеш та Індії.

Компанії, які регулярно користуються послугами контактних центрів, що передаються підрядниками, включають British Sky Broadcasting та Orange [19] у галузі телекомунікацій, Adidas у секторі спорту та дозвілля, [20] Audi у виробництві автомобілів [21] та благодійні організації, такі як RSPCA.

					<i>КНТЕУ 121 023-03.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		9

## 1.2. Сутність і основні характеристики поняття інформаційної системи

Інформаційна система - це взаємозв'язана сукупність інформаційних, технічних, програмних, математичних, організаційних, правових, ергономічних, лінгвістичних, технологічних і інших засобів, а також персоналу, призначена для збору, обробки, зберігання та видачі економічної інформації і прийняття управлінських рішень.

Властивості інформаційних систем:

1. будь-яка ІС може бути піддана аналізу, побудована і керована на основі загальних принципів побудови складних систем;
2. при побудові ІС необхідно використовувати системний підхід;
3. ІС є динамічною і розвивається системою;
4. ІС слід сприймати як систему обробки інформації, що складається з комп'ютерних і телекомунікаційних пристроїв, реалізовану на базі сучасних технологій;
5. вихідний продукцією ІС є інформація, на основі якої приймаються рішення або виробляються автоматичне виконання рутинних операцій;
6. участь людини залежить від складності системи, типів і наборів даних, ступеня формалізації вирішуваних завдань.

Процеси в інформаційній системі:

1. введення інформації з зовнішніх і внутрішніх джерел;
2. обробка вхідної інформації;
3. зберігання інформації для подальшого її використання;
4. висновок інформації в зручному для користувача вигляді;
5. зворотний зв'язок, тобто надання інформації, переробленої в даній організації, для коригування вхідної інформації.

З урахуванням сфери застосування виділяють: технічні ІС, економічні ІС, ІС в гуманітарних областях, тощо.

							Аркуш
							10
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 23-03.МР		

Типи інформаційних систем

Фактографічні і документовані інформаційні системи

Тип інформаційної системи залежить від того, чиї інтереси вона обслуговує і на якому рівні управління. За характером уявлення і логічної організації інформації, що зберігається інформаційні системи підрозділяються на фактографічні, документальні та геоінформаційні.

Фактографічні інформаційні системи накопичують і зберігають дані у вигляді безлічі екземплярів одного або декількох типів структурних елементів (інформаційних об'єктів).

Кожен з таких екземплярів або деяка їх сукупність відображають відомості з якого-небудь факту, події окремо від усіх інших відомостей і фактів.

Структура кожного типу інформаційного об'єкта складається з кінцевого набору реквізитів, що відображають основні аспекти та характеристики об'єктів даної предметної області. Комплектування інформаційної бази в фактографічних інформаційних системах включає, як правило, обов'язковий процес структуризації вхідної інформації.

Фактографічні інформаційні системи припускають задоволення інформаційних потреб безпосередньо, тобто шляхом подання споживачам самих відомостей (даних, фактів, концепцій).

У документальних (документованих) інформаційних системах одиничним елементом інформації є нерозчленованої на більш дрібні елементи документ і інформація при введенні (вхідний документ), як правило, не структурується, або структурується в обмеженому вигляді. Для вводиться документа можуть встановлюватися деякі формалізовані позиції (дата виготовлення, виконавець, тематика).

Деякі види документальних інформаційних систем забезпечують встановлення логічного взаємозв'язку документів, що вводяться -

						Аркуш
						11
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 23-03.МР	



підпорядкованість за смисловим змістом, взаємні посилання з яких-небудь критеріями, тощо.

Визначення і встановлення такого взаємозв'язку є складною многокритеріальною і багатоаспектною аналітичною задачею, яка не може бути формалізована в повній мірі.

У геоінформаційних системах дані організовані у вигляді окремих інформаційних об'єктів (з певним набором реквізитів), прив'язаних до загальної електронної топографічної основи (електронній карті).

Геоінформаційні системи застосовуються для інформаційного забезпечення в тих предметних областях, структура інформаційних об'єктів і процесів в яких має просторово-географічний компонент (маршрути транспорту, комунальне господарство).

Класифікація інформаційних систем за рівнями управління

Виділяють:

1. інформаційні системи оперативного (операційного) рівня - бухгалтерська, банківських депозитів, обробки замовлень, реєстрації квитків, виплати зарплати;
2. інформаційна система фахівців - офісна автоматизація, обробка знань (включаючи експертні системи);
3. інформаційні системи тактичного рівня (середня ланка) - моніторинг, адміністрування, контроль, прийняття рішень;
4. стратегічні інформаційні системи - формулювання цілей, стратегічне планування.

Інформаційні системи оперативного (операційного) рівня

Інформаційна система оперативного рівня підтримує фахівців-виконавців, обробляючи дані про угоди і події (рахунки, накладні, зарплата, кредити, потік сировини і матеріалів). Призначення інформаційної системи на цьому рівні - відповідати на запити про поточний стан і відслідковувати потік угод в фірмі, що відповідає оперативному управлінню.

						КНТЕУ 121 з2м-03.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			12

Щоб з цим справлятися, інформаційна система повинна бути легко доступною, безперервно діючою і надавати точну інформацію.

Завдання, цілі та джерела інформації на оперативному рівні заздалегідь визначені і у високому ступені структуровані. Рішення запрограмоване у відповідності із заданим алгоритмом.

Інформаційна система оперативного рівня є сполучною ланкою між фірмою і зовнішнім середовищем. Якщо система працює погано, то організація або не отримує інформації ззовні, або не видає інформацію. Крім того, система - це основний постачальник інформації для інших типів інформаційних систем в організації, тому що містить і оперативну, і архівну інформацію.

Інформаційні системи фахівців.

Інформаційні системи цього рівня допомагають фахівцям, що працюють з даними, підвищують продуктивність і продуктивність роботи інженерів і проектувальників. Завдання подібних інформаційних систем - інтеграція нових відомостей в організацію і допомогу в обробці паперових документів.

У міру того як індустріальне суспільство трансформується в інформаційне, продуктивність економіки все більше буде залежати від рівня розвитку цих систем. Такі системи, особливо у вигляді робочих станцій і офісних систем, найбільш швидко розвиваються сьогодні в бізнесі.

Інформаційні системи офісної автоматизації внаслідок своєї простоти і багатопрофільності активно використовуються працівниками будь-якого організаційного рівня. Найбільш часто їх застосовують працівники середньої кваліфікації: бухгалтери, секретарі, клерки.

Основна мета - обробка даних, підвищення ефективності їх роботи та спрощення канцелярської праці. Інформаційні системи офісної автоматизації пов'язують воедино працівників інформаційної сфери в різних регіонах і

					<i>КНТЕУ 121 02з-03.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		13

допомагають підтримувати зв'язок з покупцями, замовниками та іншими організаціями. Їх діяльність в основному охоплює управління документацією, комунікації, складання розкладів і т.д.

Ці системи виконують такі функції:

1. обробка текстів на комп'ютерах за допомогою різних текстових процесорів;
2. виробництво високоякісної друкованої продукції;
3. архівація документів;
4. електронні календарі і записники для ведення ділової інформації;
5. електронна і аудіопошта;
6. відео- і телеконференції.

Інформаційні системи обробки знань, в тому числі і експертні системи, вбирають в себе знання, необхідні інженерам, юристам, науковцям при розробці або створенні нового продукту. Їх робота полягає в створенні нової інформації і нового знання. Так, наприклад, існуючі спеціалізовані робочі станції з інженерного і наукового проектування дозволяють забезпечити високий рівень технічних розробок.

Інформаційні системи тактичного рівня (середня ланка)

Основні функції цих інформаційних систем:

1. порівняння поточних показників з минулими показниками;
2. складання періодичних звітів за певний час (а не видача звітів щодо поточних подій, як на оперативному рівні);
3. забезпечення доступу до архівної інформації і т.д.

Системи підтримки прийняття рішень обслуговують частково структуровані завдання, результати яких важко спрогнозувати заздалегідь (мають більш потужний аналітичний апарат з декількома моделями).

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	<i>КНТЕУ 121 023-03.МР</i>	14

Інформацію отримують з управлінських і операційних інформаційних систем. Використовують ці системи все, кому необхідно приймати рішення: менеджери, фахівці, аналітики. Наприклад, їх рекомендації можуть стати в нагоді при прийнятті рішення купувати або взяти обладнання в оренду.

### 1.3. Сутність поняття веб-сайту

Веб-сайт (Web-site), скорочено сайт (site) - сукупність загальнодоступних взаємопов'язаних веб-сторінок, які використовують одне доменне ім'я. Веб-сайти можуть створюватися і підтримуватися за окрему особу, групою, бізнесом або організацією для різних цілей. Разом все загальнодоступні веб-сайти складають всесвітню павутину.

Якщо говорити простою мовою, сайт - це адреса розташована в інтернеті, на якій знаходиться якась інформація (текст, відео, фотографії, документи, музика та інше). А інтернет - це виходить збірник таких адрес.

Або так: зараз ви перебуваєте в браузері на веб-сторінці і читаете цю інформацію, також ви можете перейти на інші сторінки і всі сторінки в сукупності (в межах даного веб-проекту) та є сайт.

Прогрес в технологіях створення веб-сайтів і безмежні можливості людської уяви визначили велике розмаїття веб-сайтів, які ми бачимо сьогодні в Інтернеті, коли задаємо питання yandex, google і іншим пошуковим системам. Однак всі вони містять деякі базові елементи, які роблять їх легко впізнаваними користувачами з точки зору функціональності - домашня сторінка (також відома як індексна), перша сторінка, яку ми бачимо при вході на веб проект; меню - що складається з гіперпосилань (посилання на основні сторінки або розділи); основний контент (вміст), область колонтитула (footer або підвал) з важливими посиланнями, інформацією про авторські права тощо.

					<i>КНТЕУ 121 023-03.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		15

За допомогою гіперпосилань відвідувач перенаправляється з одного веб-сторінки проекту на іншу.

Веб-сайтів на даний момент велика різноманітність: освітні, новинні, порнографічні, форуми, соціальні мережі, сайти електронної комерції (інтернет магазини), блоги, лендінги та інші.

#### **1.4. Процес розробки веб-сайтів, мета і методи**

Веб-розробка - це робота, пов'язана з розробкою веб-сайту для Інтернету (Всесвітня павутина) або інтрамережі (приватна мережа). [1] Веб-розробка може варіюватися від розробки простої однієї статичної сторінки простого тексту до складних веб-додатків, електронного бізнесу та послуг соціальних мереж. Більш вичерпний перелік завдань, до яких зазвичай відноситься веб-розробка, може включати веб-розробку, веб-дизайн, розробку веб-вмісту, зв'язок з клієнтом, сценарії на стороні клієнта / сервера, налаштування безпеки веб-сервера та мережі та розвиток електронної комерції.

Серед веб-спеціалістів "веб-розробка" зазвичай відноситься до основних недизайнерських аспектів побудови веб-сайтів: написання розмітки та кодування. [2] Веб-розробка може використовувати системи управління вмістом (CMS), щоб спростити зміст вмісту та отримати його базовими технічними навичками.

Для великих організацій та бізнесу групи веб-розробників можуть складатися з сотень людей (веб-розробників) і дотримуватися стандартних методів, таких як Agile, під час розробки веб-сайтів. Менші організації можуть вимагати лише одного постійного розробника або підрядника, або вторинне призначення на відповідні посади, такі як графічний дизайнер або технік інформаційних систем. Веб-розробка може бути спільним зусиллям між департаментами, а не областю призначеного відділу.

					<i>КНТЕУ 121 023-03.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		16

Існує три різновиди спеціалізації веб-розробників: розробник із інтерфейсу, розробник із задньої панелі та розробник із повним стеком. Інтернетні розробники відповідають за поведінку та візуальні ефекти, які працюють у браузері користувача, тоді як інтерфейсні розробники мають справу з серверами.

З моменту комерціалізації Інтернету веб-розробка стала зростаючою галуззю. Зростання цієї галузі відбувається за рахунок підприємств, які бажають використовувати свій веб-сайт для реклами та продажу товарів та послуг споживачам. [3]

Існує багато інструментів з відкритим кодом для веб-розробки, таких як BerkeleyDB, GlassFish, стек LAMP (Linux, Apache, MySQL, PHP) та Perl / Plack. Це дозволило звести витрати на навчання веб-розробці до мінімуму. Ще одним фактором, що сприяє зростанню галузі, стало зростання простого у використанні програмного забезпечення для веб-розробки WYSIWYG, такого як Adobe Dreamweaver, BlueGriffon та Microsoft Visual Studio. Для використання такого програмного забезпечення все ще необхідні знання мови розмітки HyperText (HTML) або мов програмування, але основи можна швидко вивчити та впровадити.

Постійно зростаючий набір інструментів та технологій допоміг розробникам створювати більш динамічні та інтерактивні веб-сайти. Крім того, веб-розробники тепер допомагають надавати додатки як веб-служби, які традиційно були доступні лише як програми на настільному комп'ютері. Це дало багато можливостей для децентралізації розповсюдження інформації та засобів масової інформації. Приклади можна побачити із зростанням хмарних сервісів, таких як Adobe Creative Cloud, Dropbox та Google Drive. Ці веб-служби дозволяють користувачам взаємодіяти з програмами з багатьох місць, замість того, щоб бути прив'язаними до певної робочої станції для свого середовища додатків.

								Аркуш
								17
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 023-03.МР			

Прикладами кардинальних перетворень у спілкуванні та комерції на чолі з веб-розробкою є електронна комерція. Інтернет-сайти аукціонів, такі як eBay, змінили спосіб споживачів знаходити та купувати товари та послуги.

Інтернет-магазини, такі як Amazon.com та Buy.com (серед багатьох інших), змінили досвід покупок та торгівлі для багатьох споживачів. Іншим прикладом трансформаційного спілкування, керованого веб-розробкою, є блог.

Веб-програми, такі як WordPress та Movable Type, створили середовища для ведення блогів для окремих веб-сайтів. Поширене використання систем управління вмістом з відкритим кодом та систем управління корпоративним контентом розширило вплив веб-розробки на взаємодію та спілкування в Інтернеті.

Розробка веб-сайтів також вплинула на особисті мережі та маркетинг. Веб-сайти більше не є просто інструментами для роботи чи комерції, а ширше служать для спілкування та соціальних мереж. Такі веб-сайти, як Facebook та Twitter, надають користувачам платформу для спілкування, а організації - більш особистий та інтерактивний спосіб залучення громадськості.

Веб-розробка враховує багато міркувань безпеки, таких як перевірка помилок при введенні даних через форми, фільтрація вихідних даних та шифрування. Шкідливі практики, такі як ін'єкція SQL, можуть виконуватися користувачами з ненавмисними намірами, але лише з примітивними знаннями про веб-розробку в цілому. Сценарії можна використовувати для використання веб-сайтів, надаючи несанкціонований доступ зловмисним користувачам, які намагаються збирати таку інформацію, як адреси електронної пошти, паролі та захищений вміст, наприклад, номери кредитних карток.

					<i>КНТЕУ 121 023-03.МР</i>	Аркуш
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		18

Деякі з них залежать від середовища сервера, на якому запущена мова сценаріїв, наприклад ASP, JSP, PHP, Python, Perl або Ruby, і тому не обов'язково підтримувати самого веб-розробника. Однак рекомендується жорстке тестування веб-додатків перед публічним випуском, щоб запобігти подібним подвигам. Якщо на веб-сайті є якась контактна форма, вона повинна включати в неї поле captcha, яке не дозволяє комп'ютерним програмам автоматично заповнювати форми, а також розсилати пошту.

Оскільки нові веб-програми виявляють нові діри в безпеці навіть після тестування та запуску, оновлення виправлень безпеки часто трапляються для широко використовуваних програм.

Часто робота веб-розробників - постійно оновлювати програми, коли випускаються виправлення безпеки та виявляються нові проблеми безпеки.

Захист веб-сервера від вторгнень часто називають зміцненням порту сервера. Багато технологій застосовуються для захисту інформації в Інтернеті, коли вона передається з одного місця в інше. Наприклад, сертифікати TLS (або "сертифікати SSL") видаються органами сертифікації для запобігання шахрайству в Інтернеті. Багато розробників часто використовують різні форми шифрування при передачі та зберіганні конфіденційної інформації. Базове розуміння проблем безпеки інформаційних технологій часто є частиною знань веб-розробника.

### **Висновки до розділу 1**

Перший розділ присвячений аналізу теоретичної бази предметної області обраної тематики, який націлений на отримання загального розуміння обраної теми і в подальшому стане основою для вибору інструментарію розробки, проектування та реалізації програмного продукту.

В першому розділі проаналізовано поняття call-центру, інформаційної системи, веб-сайту та процес створення веб сайтів, як основні теоретичні модулі предметної області створюваного програмного продукту.

						КНТЕУ 121 023-03.МР	Аркуш
							19
Зм.	Аркуш	№ докум	Підпис	Дата			



## РОЗДІЛ 2

### ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ, ОПИС ТА ОБГРУНТУВАННЯ ЗАСОБІВ РЕАЛІЗАЦІЇ

#### 2.1. Постановка задачі розробки

Основна задача даної роботи полягає в проектуванні і реалізації системи стратегічного аналізу діяльності call-центру. Основна задача розроблюваної системи полягає в оцінці експертами якості роботи операторів по певному набору критеріїв, з метою покращення якості роботи операторів. Загальний функціонал програмного рішення, який необхідно реалізувати, включає наступні пункти:

1. механізми реєстрації і авторизації користувачів;
2. розподілення ролей користувачів;
3. можливість створення нових критеріїв оцінки роботи операторів;
4. можливість експертів оцінювати роботу того чи іншого оператора;
5. можливість керівництва вносити корективи і коментарі в результат оцінки роботи операторів;
6. формування результату оцінювання роботи операторів на базі оцінок експертів;
7. можливість операторів переглядати результати;
8. формування повноцінного результату оцінювання роботи операторів з набором рекомендацій щодо покращення якості роботи операторів.

#### 2.2. Вибір мови програмування

Python - інтерпретована мова програмування високого рівня та загального призначення. Філософія дизайну Python наголошує на

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>КНТЕУ 121 023-03.МР</i>			
Зав. каф.		Криворучко О.В.		24.05.21	<i>Проектування ПЗ системи стратегічного аналізу діяльності call-центру</i>	Стадія	Аркуш	Аркушів
Керівник		Рзаєва С.Л.		24.05.21		P2	20	48
Гарант		Токар В.В.		24.05.21		Факультет інформаційних технологій 2 курс, 23 група		
Розробив		Бойко А.А.		24.05.21				
					<i>Постановка задачі розробки, опис та обґрунтування</i>			

читабельності коду завдяки помітному використанню значних відступів. Його мовні конструкції та об'єктно-орієнтований підхід мають на меті допомогти програмістам написати чіткий логічний код для малих та великих проектів.

Python динамічно набирається та збирається сміття. Він підтримує декілька парадигм програмування, включаючи структуроване (зокрема, процедурне), об'єктно-орієнтоване та функціональне програмування. Python часто описують як мову, що включає батареї, завдяки своїй повній стандартній бібліотеці.

Python стабільно входить до числа найпопулярніших мов програмування.

Python - мова програмування з багатьма парадигмами. Об'єктно-орієнтоване програмування та структуроване програмування повністю підтримуються, і багато його функцій підтримують функціональне програмування та аспектно-орієнтоване програмування (в тому числі метапрограмуванням та метаоб'єктами (магічні методи)). Багато інших парадигм підтримуються за допомогою розширень, включаючи дизайн за контрактом та логічне програмування.

Python використовує динамічне введення тексту та комбінацію підрахунку посилань та збирач сміття, що визначає цикл, для управління пам'яттю. Він також має динамічну роздільну здатність імен (пізніє прив'язування), яка зв'язує імена методів та змінних під час виконання програми.

Дизайн Python пропонує певну підтримку функціонального програмування за традицією Lisp. Він має функції фільтрування, картографування та зменшення; перелічити розуміння, словники, набори та вирази генераторів. Стандартна бібліотека має два модулі (itertools та functools), що реалізують функціональні інструменти, запозичені у Haskell та Standard ML.

Основна філософія мови узагальнена в документі Zen of Python (PEP 20), який включає такі афоризми, як:

- Красиве - краще, ніж потворне.

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		21

- Явне краще, ніж неявне.
- Просте - краще, ніж складне.
- Складний - це краще, ніж складний.
- Підрахунок читабельності.

Замість того, щоб усі його функціональні можливості були вбудовані в його ядро, Python був розроблений таким чином, щоб бути дуже розширюваним (з модулями). Ця компактна модульність зробила його особливо популярним як засіб додавання програмованих інтерфейсів до існуючих додатків. Бачення Ван Россума щодо малої основної мови з великою стандартною бібліотекою та легко розширюваним перекладачем впливало з його розчарувань у ABC, який підтримував протилежний підхід. Python прагне до більш простого, менш захащеного синтаксису та граматики, одночасно надаючи розробникам можливість вибору методології кодування. На відміну від девізу Perl "існує більше ніж один спосіб зробити це", Python охоплює "повинен бути один - і бажано лише один - очевидний спосіб зробити це" філософія дизайну. Алекс Мартеллі, співробітник Фонду програмного забезпечення Python і автор книги про Python, пише, що "Описувати щось як" розумне "не вважається компліментом у культурі Python".

Розробники Python прагнуть уникнути передчасної оптимізації та відкидають виправлення для некритичних частин еталонної реалізації CPython, які пропонують незначне збільшення швидкості ціною ясності. Коли швидкість важлива, програміст Python може переміщати критично важливі для часу функції до модулів розширення, написаних мовами, такими як C, або використовувати PyPy, своєчасний компілятор. Також доступний Cython, який перекладає скрипт Python на C і здійснює прямі виклики API рівня C в інтерпретатор Python.

Важливою метою розробників Python є підтримка його задоволення від використання. Це відображається в назві мови - данина поваги британській гумористичній групі Монті Пайтон - і в іноді грайливих підходах до

						Аркуш
						22
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 023-03.МР	

навчальних посібників та довідкових матеріалів, таких як приклади, що стосуються спаму та яєць (із відомого етюдю Монті Пайтона) стандартних foo and bar.

Поширеним неологізмом у спільноті Python є пітонічний, який може мати широкий діапазон значень, пов'язаних зі стилем програми. Сказати, що код пітонічний, означає сказати, що він добре використовує ідіоми Python, що він природний або демонструє вільну мову, що відповідає мінімалістичній філософії Python та наголосу на читабельності. На відміну від цього, код, який важко зрозуміти або читається як груба транскрипція з іншої мови програмування, називається непітонічним.

Користувачів та шанувальників Python, особливо тих, хто вважається обізнаним чи досвідченим, часто називають Pythonistas. Python має бути легкочитабельною мовою. Його форматування візуально не захаращене, і в ньому часто використовуються англійські ключові слова, де інші мови використовують розділові знаки. На відміну від багатьох інших мов, він не використовує фігурні дужки для розмежування блоків, а крапка з комою після операторів дозволена, але рідко, якщо взагалі використовується. Він має менше синтаксичних винятків та особливих випадків, ніж C або Pascal.

Python використовує пробіли, а не фігурні дужки або ключові слова, для відмежування блоків. Збільшення відступу відбувається після певних тверджень; зменшення відступу означає кінець поточного блоку. Таким чином, візуальна структура програми точно відображає семантичну структуру програми. Цю функцію іноді називають правилом "поза стороною", яке мають деякі інші мови, але в більшості мов відступ не має семантичного значення. Рекомендований розмір відступу - чотири пробіли.

Python може служити мовою сценаріїв для веб-додатків, наприклад, через mod\_wsgi для веб-сервера Apache. Завдяки інтерфейсу шлюзу веб-сервера для полегшення роботи цих програм розроблено стандартний API.

						Аркуш
						23
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 023-03.MP	

Веб-фреймворки, такі як Django, Pylons, Pyramid, TurboGears, web2py, Tornado, Flask, Bottle і Zope, підтримують розробників у розробці та обслуговуванні складних додатків. Pyjs та IronPython можуть бути використані для розробки клієнтської частини програм на базі Ajax. SQLAlchemy може використовуватися як перетворювач даних у реляційну базу даних. Twisted - це фреймворк для програмування зв'язку між комп'ютерами, який використовується (наприклад) Dropbox.

Такі бібліотеки, як NumPy, SciPy та Matplotlib, дозволяють ефективно використовувати Python у наукових обчисленнях, а спеціалізовані бібліотеки, такі як Biopython та Astropy, забезпечують функціональність, що залежить від домену. SageMath - математичне програмне забезпечення з інтерфейсом ноутбука, програмоване на Python: його бібліотека охоплює багато аспектів математики, включаючи алгебру, комбінаторику, числову математику, теорію чисел і числення. OpenCV має палітурні прив'язки з багатим набором функцій для комп'ютерного зору та обробки зображень.

Python зазвичай використовується в проєктах штучного інтелекту та машинному навчанні за допомогою таких бібліотек, як TensorFlow, Keras, Pytorch та Scikit-learn. Як мова сценаріїв з модульною архітектурою, простим синтаксисом та інструментами обробки багатого тексту, Python часто використовується для обробки природної мови. Python успішно вбудований у багато програмних продуктів як мову сценаріїв, включаючи програмне забезпечення методом скінченних елементів, таке як Abaqus, 3D-параметричний моделер, такий як FreeCAD, пакети 3D-анімації, такі як 3ds Max, Blender, Cinema 4D, Lightwave, Houdini, Maya, modo, MotionBuilder, Softimage, композитор візуальних ефектів Nuke, програми для 2D-зображень, такі як GIMP, Inkscape, Scribus і Paint Shop Pro, та музичні нотатні програми, такі як автори та капели. Налаштовувач GNU використовує Python як гарний принтер для показу складних структур, таких як контейнери C++.

					<i>КНТЕУ 121 02з-03.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		24

Esri пропагує Python як найкращий вибір для написання сценаріїв в ArcGIS. Він також використовувався в декількох відеоіграх і був прийнятий як перша з трьох доступних мов програмування в Google App Engine, інші дві - Java і Go.

Багато операційних систем включають Python як стандартний компонент. Він постачається з більшістю дистрибутивів Linux, AmigaOS 4 (з використанням Python 2.7), FreeBSD (як пакет), NetBSD, OpenBSD (як пакет) та macOS і може використовуватися з командного рядка (терміналу). Багато дистрибутивів Linux використовують інсталятори, написані на Python: Ubuntu використовує інсталятор Ubiquity, тоді як Red Hat Linux і Fedora використовують інсталятор Anaconda. Gentoo Linux використовує Python у своїй системі управління пакунками Portage.

Python широко використовується в галузі інформаційної безпеки, в тому числі в розробці експлоїтів. Більшість програм Sugar для одного ноутбука на дитину XO, які зараз розробляються в Sugar Labs, написані на Python.

Проект одноплатної комп'ютерної програми Raspberry Pi прийняв Python як основну мову програмування користувачів.

LibreOffice включає Python і має намір замінити Java на Python. Його постачальник сценаріїв Python є основною функцією з версії 4.0 від 7 лютого 2013 року.

Виходячи зі сфери використання і величезного функціоналу мови Python, який був описаний вище, можна зробити висновок про його мультифункціональність. Саме через це дана мова програмування була обрана для виконання даної роботи.

						Аркуш
						25
Зм.	Аркуш	№ докум	Підпис	Дата	<i>КНТЕУ 121 02з-03.МР</i>	

### 2.3. Вибір середовища розробки інформаційної системи

PyCharm – це інтегроване середовище розробки (IDE), що використовується в комп'ютерному програмуванні, зокрема для мови Python. Він розроблений чеською компанією JetBrains. Він забезпечує аналіз коду, графічний налагоджувач, інтегрований тестер одиниць, інтеграцію із системами контролю версій (VCSes), а також підтримує веб-розробку з Django, а також аналіз даних з Anaconda.

PyCharm є крос-платформним, з версіями для Windows, macOS та Linux. Видання Community випускається під ліцензією Apache, а також існує Professional Edition з додатковими функціями - випущено під власною ліцензією.

### 2.4. Вибір системи керування базами даних

Бази даних використовуються для структурованого зберігання даних, а SQLite - це популярний формат баз даних, що з'являється у багатьох мобільних системах, а також традиційних операційних системах. SQLite - це технологічна бібліотека, яка реалізує самостійний, безсерверний, механізм баз даних SQL з нульовою конфігурацією. Код SQLite знаходиться у відкритому доступі і, отже, безкоштовний для будь-яких цілей, комерційних або приватних.

SQLite - це вбудований механізм баз даних SQL. На відміну від більшості інших баз даних SQL, SQLite не має окремого серверного процесу. SQLite читає та пише безпосередньо на звичайні файли диска. Повна база даних SQL з кількома таблицями, індексами, тригерами та поданнями міститься в одному дисковому файлі. Як правило, SQLite працює швидше, чим більше пам'яті ви йому надаєте. Тим не менше, продуктивність, як правило, досить хороша навіть у середовищах з низьким обсягом пам'яті. Залежно від того, як він використовується, SQLite може бути швидшим, ніж прямий ввід / вивід файлової системи.

						КНТЕУ 121 02з-03.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			26

Більшість вихідних кодів SQLite присвячені виключно тестуванню та верифікації. Автоматизований набір тестів запускає мільйони і мільйони тестових випадків, що включають сотні мільйонів окремих операторів SQL, і забезпечує 100% покриття тестуванням філій. SQLite витончено реагує на помилки розподілу пам'яті та помилки вводу-виводу диска. Все це перевіряється автоматизованими тестами за допомогою спеціальних тестових джгутів, які імітують відмови системи. Звичайно, навіть при всьому цьому тестуванні все ще існують помилки. Але на відміну від деяких подібних проектів (особливо комерційних конкурентів), SQLite відкрито та чесно ставиться до всіх помилок і надає списки помилок та щохвилинні хронології змін коду.

Отож, для наглядного огляду переваг взято одного з конкурентів SQLite, а саме SQL Server, і в свою чергу було здійснено їхнє коротке порівняння:

- Модель ціноутворення:
  - SQLite безкоштовний;
- Серверні операційні системи:
  - SQLite не потребує сервера;
  - SQL Server працює на Linux та Windows;
- API та інші методи доступу:
  - SQLite підтримує такі драйвери:
    - ADO.NET;
    - JDBC;
    - ODBC;
  - SQL Server підтримує:
    - OLE DB;
    - ADO.NET;
    - JDBC;
    - ODBC;
    - Табличний потік даних (TDS);

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	<i>КНТЕУ 121 023-03.МР</i>	27



- Підтримувані мови програмування:
  - SQL Server підтримує такі мови програмування (C#, C++, Delphi, Go, Java, JavaScript (Node.js), PHP, Python, R, Ruby, Visual Basic);
  - SQLite підтримує майже будь-які мови програмування, про які ви можете подумати (Actionscript, Ada, Basic, C, C#, C++, D, Delphi, Forth, Fortran, Haskell, Java, JavaScript, Lisp, Lua, MatLab, Objective-C, OCaml, Perl, PHP, PL/SQL, Python, R, Ruby, Scala, Scheme, Smalltalk, Tcl);
- Підтримка збереженої процедури:
  - SQLite не підтримує збережену процедуру;
  - SQL Server має його з мовами Transact SQL та .NET;
- Методи розділення:
  - SQLite не підтримує;
  - У SQL Server таблиці можуть розподілятися між кількома файлами (горизонтальне розділення);
- Методи реплікації:
  - SQLite не підтримує;
  - SQL Server підтримує;
- Контроль доступу користувачів:
  - SQLite не має концепції контролю доступу користувачів.

SQL Server має чіткі права доступу відповідно до стандарту SQL.

Виходячи з усіх описаних вище переваг SQLite та факту первинної наявності бази даних SQLite в проектах Django, основною СКБД для розроблення даної інформаційної системи обрано саме SQLite.

					<i>КНТЕУ 121 02з-03.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		28

## 2.5. Вибір фреймворку веб-розробки для мови Python

Django - це безкоштовний веб-фреймворк на основі Python, що відповідає архітектурному зразку model-template-views (MTV). Він підтримується Django Software Foundation (DSF), американською незалежною організацією, створеною як некомерційна організація 501 (c) (3).

Основна мета Django - полегшити створення складних веб-сайтів, керованих базами даних. Структура підкреслює багаторазовість та "підключеність" компонентів, менше коду, низьку зв'язок, швидкий розвиток та принцип "не повторюватися". Python використовується всюди, навіть для налаштувань, файлів та моделей даних. Django також надає додатковий адміністративний інтерфейс створення, читання, оновлення та видалення, який генерується динамічно за допомогою самоаналізу та налаштовується за допомогою моделей адміністратора.

Django також надає додатковий адміністративний інтерфейс створення, читання, оновлення та видалення, який генерується динамічно за допомогою самоаналізу та налаштовується за допомогою моделей адміністратора.

Деякі добре відомі сайти, які використовують Django, включають PBS, Instagram, Mozilla, The Washington Times, Disqus, Bitbucket, і Nextdoor.

## 2.6. Висновок до розділу 2

В ході написання другого розділу роботи було проведено аналіз існуючих інструментальних засобів вирішення поставленої задачі. Обрано і проаналізовано мову програмування, середовище розробки і систему керування базами даних, веб-фреймворк Django, знання про які в подальшому допоможуть правильно побудувати підхід до розроблення моделі програмного забезпечення.

						КНТЕУ 121 023-03.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			29

## РОЗДІЛ 3

### ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

#### 3.1. Структурна схема інформаційної системи

Структуру інформаційної системи складає сукупність окремих її частин - підсистем. Підсистема - це частина системи, яка виділена за певною ознакою. Тому структура будь-якої інформаційної системи може бути представлена як сукупність підсистем, що забезпечують інформаційне, технічне, програмне, організаційне забезпечення.

На рис. 3.1 зображена структурна схема, яка описує взаємодію між модулями розробленої системи.

На схемі показано 3 основних модулі програмного забезпечення, яке розроблюється, а саме:

- графічний інтерфейс користувача, який відповідає за взаємодію користувача з програмним рішенням і несе в собі функціонал передачі запитів від користувача до бекенду і від бекенду до користувача;
- бекенд, який містить в собі весь функціонал програми по відправці запитів до бази даних, отриманню відповідей від бази даних, створення придатного для людського ока зовнішнього вигляду отриманих від бази даних результатів;
- база даних, яка містить в собі усі записи стосовно користувачів, оцінювання роботи операторів, критеріїв оцінювання роботи операторів, тощо.

					<i>КНТЕУ 121 023-03.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		21.06.21	<i>Проектування ПЗ системи стратегічного аналізу діяльності call-центру</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Рзасва С.Л.		21.06.21		<i>Р3</i>	<i>30</i>	<i>50</i>
Гарант		Токар В.В.		21.06.21		<i>Факультет інформаційних технологій 2 курс, 23 група</i>		
Розробив		Бойко А.А.		21.06.21				

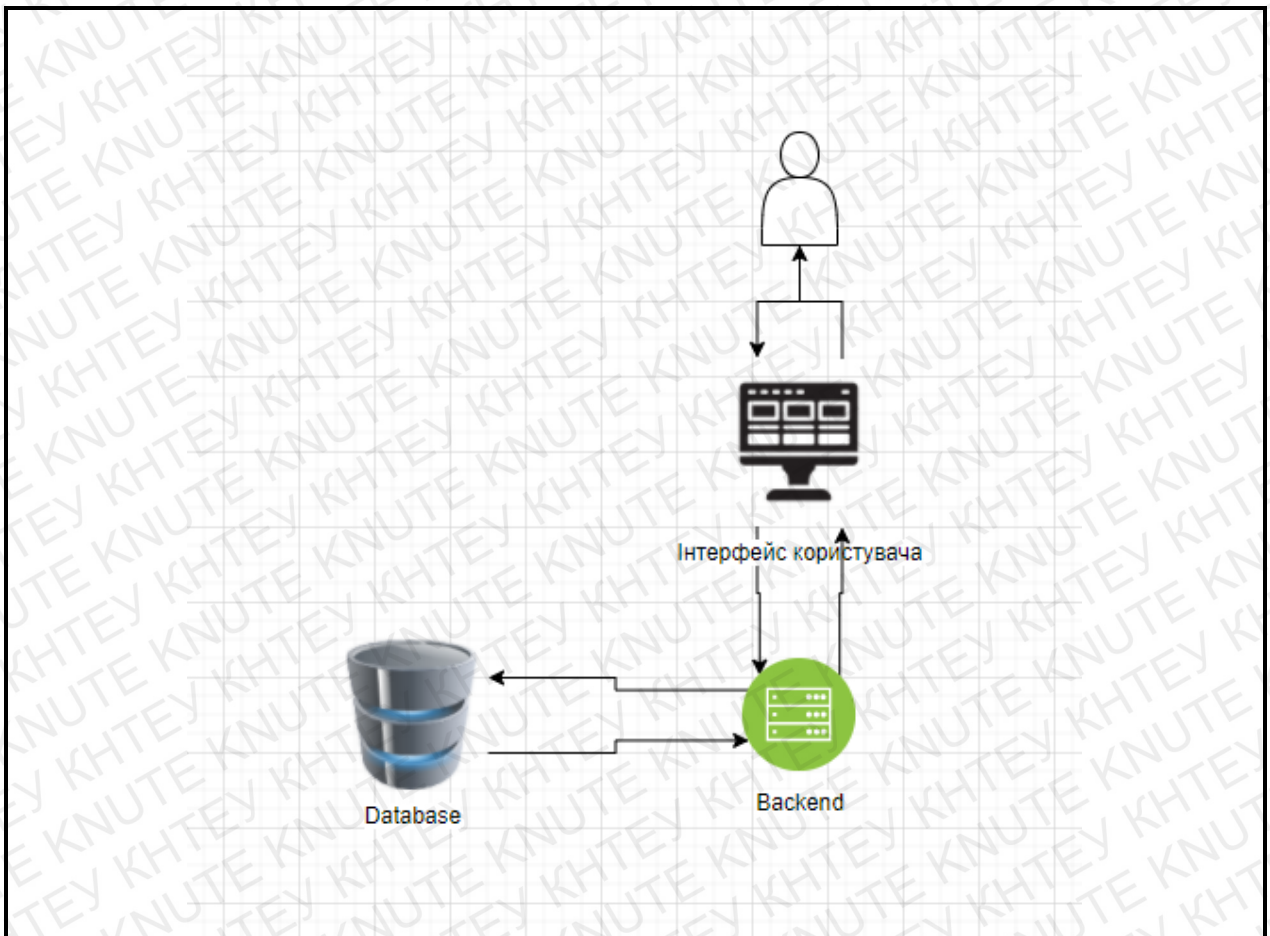


Рис. 3.1 — Структурна схема розробленої інформаційної системи

### 3.2. Моделювання діаграми варіантів використання інформаційної системи

Діаграма використання найпростіша - це представлення взаємодії користувача із системою, яка показує взаємозв'язок між користувачем та різними випадками використання, в яких користувач бере участь. Діаграма випадків використання може ідентифікувати різні типи користувачів системи та різні випадки використання, і часто вона супроводжується також іншими типами діаграм. Варіанти використання представлені кругами або еліпсами. Незважаючи на те, що сам випадок використання може детально вивчити кожну можливість, діаграма прикладів використання може допомогти забезпечити огляд системи на більш високому рівні. Раніше вже було сказано, що "схеми використання - це принципи вашої системи".

Через їх спрощений характер, схеми використання можуть бути хорошим інструментом комунікації для зацікавлених сторін. Креслення намагаються імітувати реальний світ і дають зацікавленій стороні уявлення про те, як буде розроблена система.

Метою діаграми використання є відображення динамічного аспекту системи. Додаткові схеми та документація можуть бути використані для забезпечення повного функціонального та технічного уявлення про систему. Вони забезпечують спрощене та графічне представлення того, що система насправді повинна робити.

Елементи:

- рамки системи (англ. system border) - прямокутник із назвою у верхніх частинах та еліпсами (прецедентами) всередині. Часто може бути опущено без корисної інформації про полезну інформацію,
- актор (англ. actor) - стилізований людський персонаж, обзначаючий набір ролей користувача (розуміється в широкому змісті: людина, зовнішня сутність, клас, інша система), взаємодіючого з деякою сутністю (системною, підсистемою, класом). Актори не можуть бути пов'язані між собою з іншим (за вимкнення відносин щодо обробки / дослідження),
- прецедент - еліпс із надписом, що означає виконувану систематичну дію (може включати можливі варіанти), що призводить до спостерігаємих акторами результатів.

Надпис може бути ім'ям або описом (з точки зору актора) того, "що" робить система (а не "як"). Ім'я прецедента зв'язано з неперервним (атомарним) сценарієм - конкретною послідовністю дій, ілюструючою поведінку. Під час сценарію актори обмінюються із систематичними повідомленнями. Сценарій може бути приведений на діаграмі прецедентів у відео UML-коментарі. З одним прецедентом може бути пов'язано кілька різних сценаріїв.

					КНТЕУ 121 023-03.МР	Аркуш
						32
Зм.	Аркуш	№ докум	Підпис	Дата		

На рисунку 3.2 зображено діаграму варіантів використання, яка описує можливі дії користувача в системі.

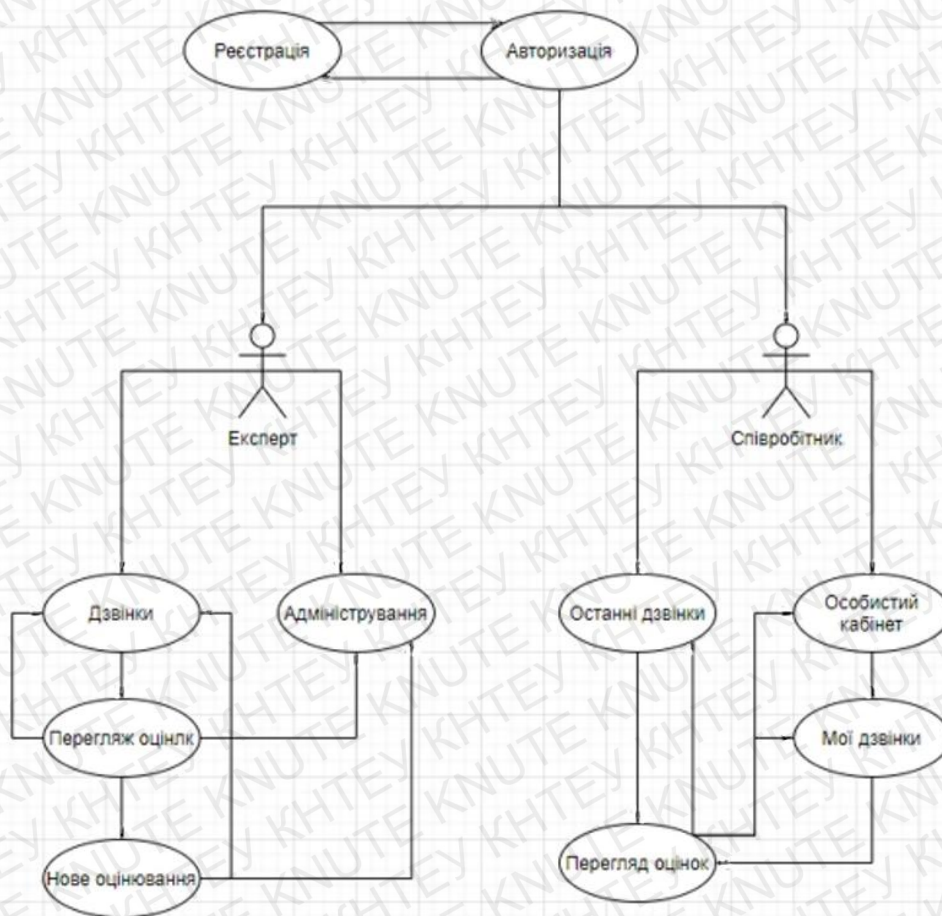


Рис. 3.2 – Схема процесу розроблення інформаційної системи стратегічного аналізу call-центру

З діаграми видно, що в системі присутні дві ролі користувачів, а саме експерт і співробітник. Експерт має доступ до створення нових оцінювань і панелі адміністратора, яка дає можливість створювати в системі нові дзвінки та вносити зміни в існуючі дзвінки і оцінювання. Окрім того експерт має доступ до перегляду інформації щодо оцінювань дзвінків іншими експертами.

Співробітник, в свою чергу, має доступ до перегляду усіх дзвінків та їх оцінок або перегляду лише особистих дзвінків через функцію особистого кабінету.

### 3.3. Висновок до розділу 3

Третій розділ даної роботи присвячено моделюванню майбутньої системи стратегічного аналізу call-центру. Моделювання виконувалося за допомогою самописної структурної діаграми, яка показує внутрішню модульну будову системи і запитів між модулями і не має аналогів. Окрім цього було проведено аналіз варіантів використання програмного продукту, який покликаний отримати загальне розуміння про майбутній функціонал програмного забезпечення стратегічного аналізу call-центру. Усі отримані дані є основою для розробки програмного забезпечення системи стратегічного аналізу call-центру.

					<i>КНТЕУ 121 023-03.MP</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		34

## РОЗДІЛ 4

### РЕАЛІЗАЦІЯ СИСТЕМИ СТРАТЕГІЧНОГО АНАЛІЗУ ДІЯЛЬНОСТІ CALL-ЦЕНТРУ

#### 4.1. Модель діаграми класів

У програмній інженерії діаграма класів в Уніфікованій мові моделювання (UML) - це тип статичної структурної діаграми, що описує структуру системи, показуючи класи системи, їх атрибути, операції (або методи) та взаємозв'язки між об'єктами.

Діаграма класів є основним будівельним елементом об'єктно-орієнтованого моделювання. Він використовується для загального концептуального моделювання структури програми та для детального моделювання переведення моделей у програмовий код. Діаграми класів також можуть бути використані для моделювання даних. Класи на діаграмі класів представляють як основні елементи, взаємодії в програмі, так і класи, що програмуються.

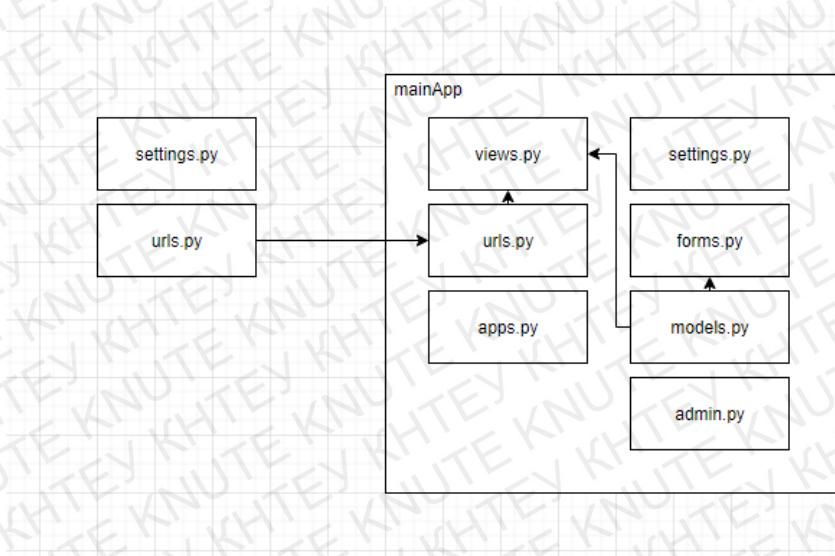


Рис. 4.1 — Модель діаграми класів

					<i>КНТЕУ 121 023-03.МР</i>		
Зм.	Аркуш	№ докум.	Підпис	Дата	Проектування ПЗ системи стратегічного аналізу діяльності call-центру  Реалізація системи стратегічного аналізу діяльності call-центру call -центру		
Зав. каф.		Криворучко О.В.		20.09.21			
Керівник		Рзасва С.Л.		20.09.21			
Гарант		Токар В.В.		20.09.21			
Розробив		Бойко А.А.		20.09.21			
					Стадія	Аркуш	Аркушіів
					РЗ	35	48
					Факультет інформаційних технологій 2 курс, 23 група		



На рисунку 4.1 зображена модель діаграми класів програмного продукту, яка повністю описує внутрішню будову системи з точки зору архітектури.

#### 4.2. Розробка графічного інтерфейсу користувача

Графічний інтерфейс користувача складається з окремих сторінок веб-ресурсу, розділених по ролям користувачів, а саме:

- сторінка авторизації (рис. 4.2);
- сторінка реєстрації (рис. 4.3);
- сторінки співробітника:
  - сторінка оцінювань (рис. 4.4);
  - сторінка особистого кабінету;
  - сторінка перегляду оцінок;
- сторінки експерта:
  - сторінка оцінювань;
  - сторінка перегляду оцінок (рис. 4.5);
  - сторінка створення нової оцінки (рис. 4.6);
  - сторінка панелі адміністратора (рис. 4.7).

На рисунку 4.2 зображено форму авторизації користувача. Дана форма єдина і для експерта, і для оператора, проте перенаправляє кожного на відповідну сторінку веб-ресурсу. Форма містить 2 поля для вводу пароля і логіну та кнопку для входу.

					<i>КНТЕУ 121 023-03.МР</i>	Аркуш
						36
Зм.	Аркуш	№ докум	Підпис	Дата		

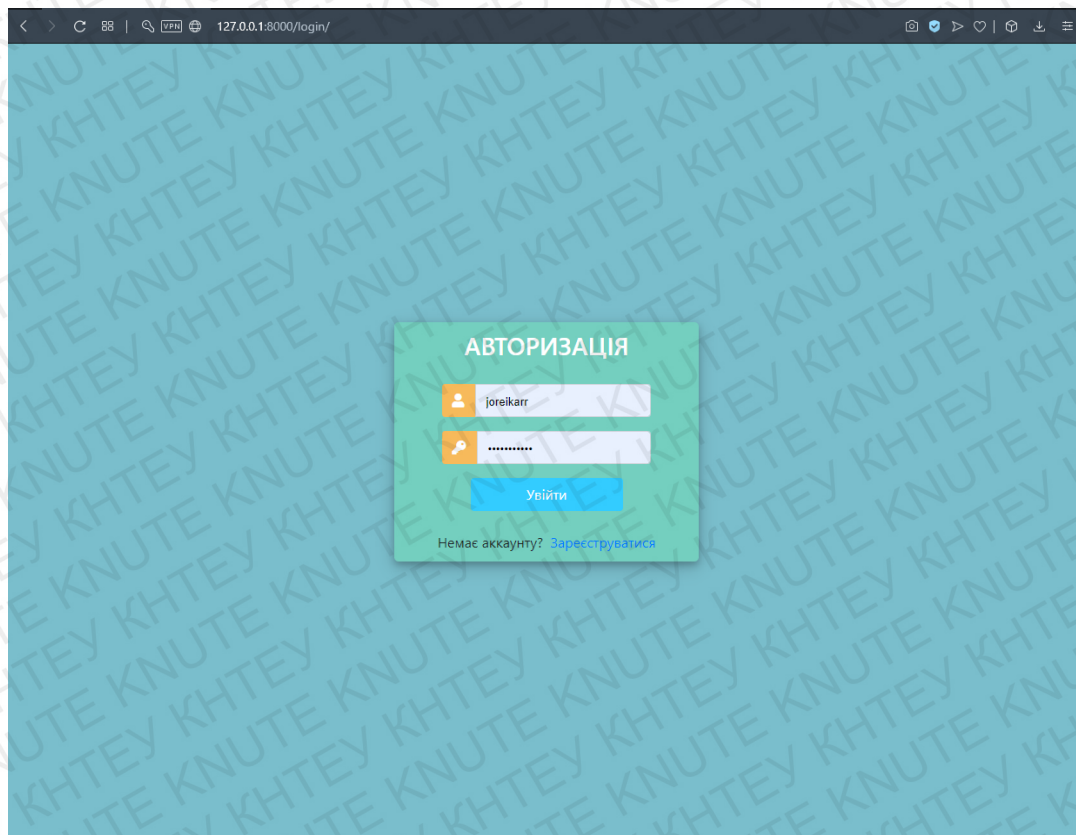


Рис. 4.2 – Сторінка авторизації інформаційної системи

На рисунку 4.3 зображено сторінку реєстрації співробітників, дана сторінка підходить тільки для реєстрації співробітників, оскільки реєстрація акаунтів експертів проводиться існуючим експертом через панель адміністратора. Форма реєстрації містить в собі 4 поля для вводу логіну, пошти, паролю та перевірки паролю, кнопку реєстрації та зворотне посилання на сторінку авторизації.

						Аркуш
					<i>КНТЕУ 121 023-03.МР</i>	37
Зм.	Аркуш	№ докум	Підпис	Дата		

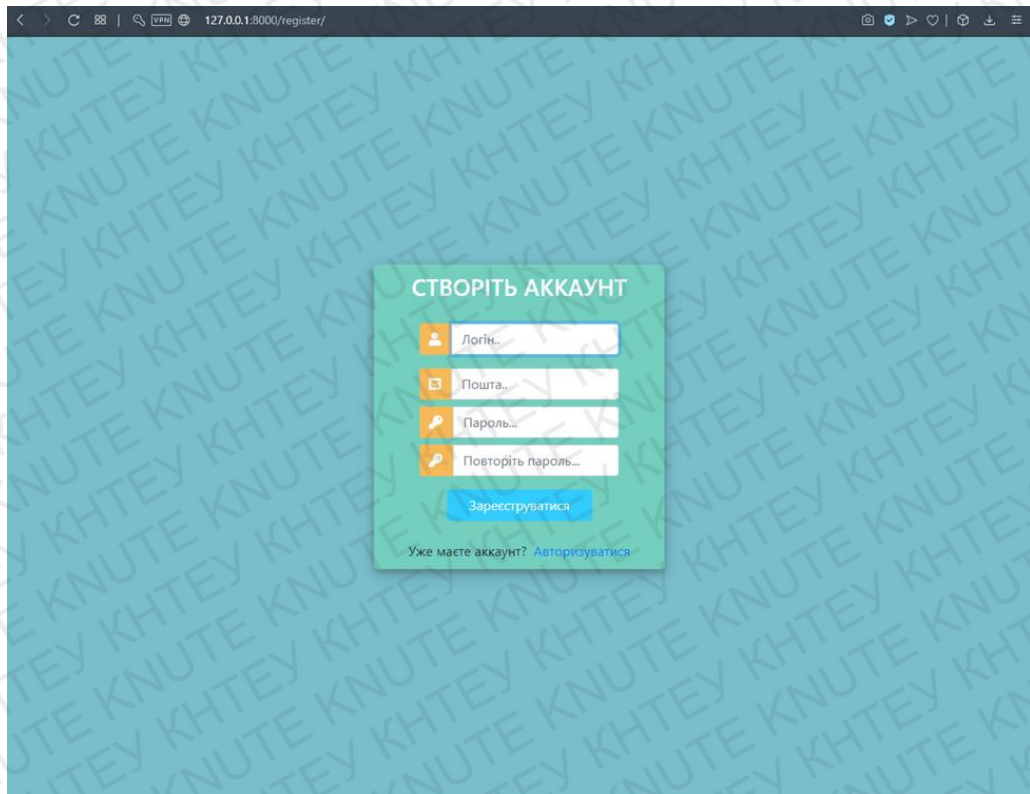


Рис. 4.3 – Сторінка для реєстрації користувачів в інформаційній системі

На рисунку 4.4 зображено сторінку оцінювань. Дана сторінка є аналогічною для обох ролей користувачів, за виключенням навігаційного меню, яке містить інші пункти для експертів. Сторінка містить список усіх відкритих оцінювань по дзвінкам. У списку виводиться логін оператора, дата та час проведення дзвінка. При натисканні на посилання користувач переходить на сторінку перегляду оцінок відповідного дзвінка. Сторінка кабінету оператора аналогічна даній, окрім того, що на сторінці кабінету виведено лише дзвінки поточного користувача і через це в списку дзвінків не відображується логін оператора.

					<i>КНТЕУ 121 023-03.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		38

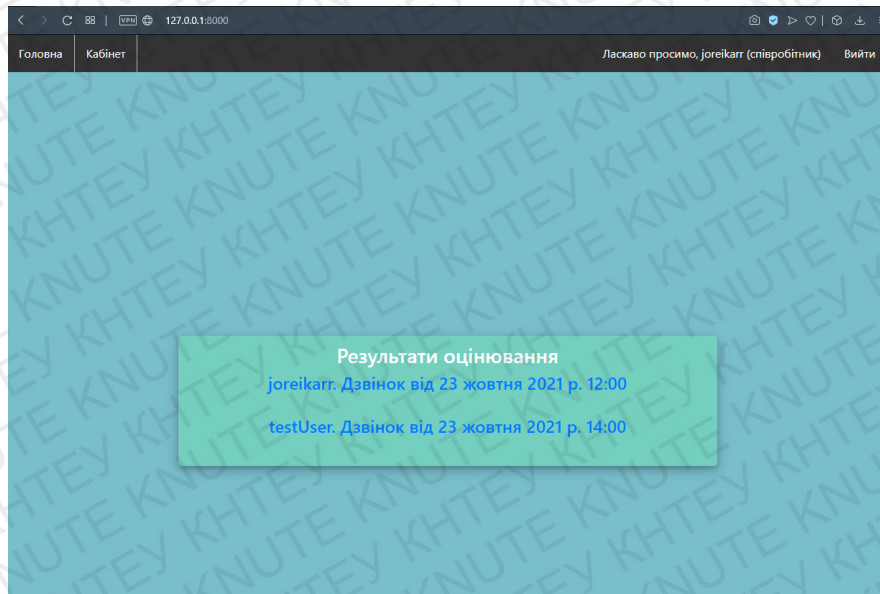


Рис. 4.4 — Сторінка оцінювань

На рисунку 4.5 зображено сторінку показу оцінок конкретного дзвінка. Сторінка представляє собою список усіх оцінок по даному дзвінку, виведених у стовпчик і розділених по експертам. Сторінка перегляду оцінок для операторів аналогічна даній, за виключенням наявності на сторінці експертів посилання на сторінку створення нової оцінки, що знаходиться в нижній частині сторінки.

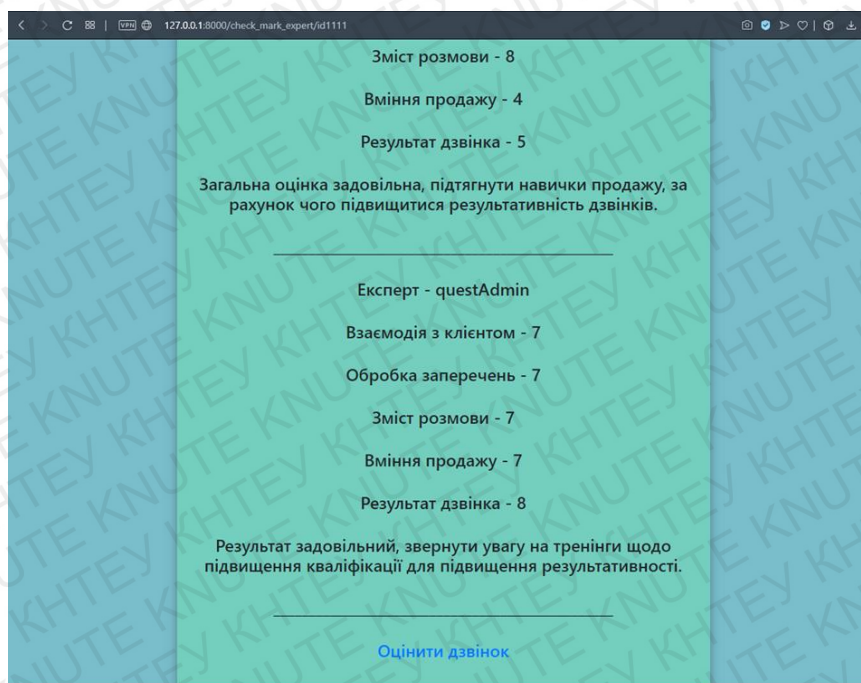


Рис. 4.5 – Сторінка перегляду оцінок

					<i>KHTEU 121 023-03.MP</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		39

На рисунку 4.6 зображено сторінку створення нової оцінки для дзвінка. Сторінка представляє собою набір полів, призначених для введення усіх необхідних даних, що стосуються оцінки дзвінка.

The screenshot shows a web browser window with the URL 127.0.0.1:8000/new\_mark/id1111. The page title is 'Нова оцінка'. The form contains the following elements:

- Експерт:** A dropdown menu with 'admin' selected.
- Дзвінок:** A text input field containing 'joreika@gmail.com, Дзвінок від: 2021-10-23 12:00:00'.
- Взаємодія з клієнтом:** A text input field.
- Зміст розмови:** A text input field.
- Вміння продажу:** A text input field.
- Вміння продаж:** A text input field.
- Результат дзвінка:** A text input field.
- Оцінка и рекомендації:** A large text area for notes.
- Додати оцінку:** A button at the bottom of the form.

Рис. 4.6 – Сторінка створення нового оцінювання

На рисунку 4.7 зображено сторінку панелі адміністратора, яка дозволяє експерту взаємодіяти з базою даних напряму через зручний інтерфейс користувача.

						Аркуш
						40
Зм.	Аркуш	№ докум	Підпис	Дата		

*КНТЕУ 121 023-03.МР*

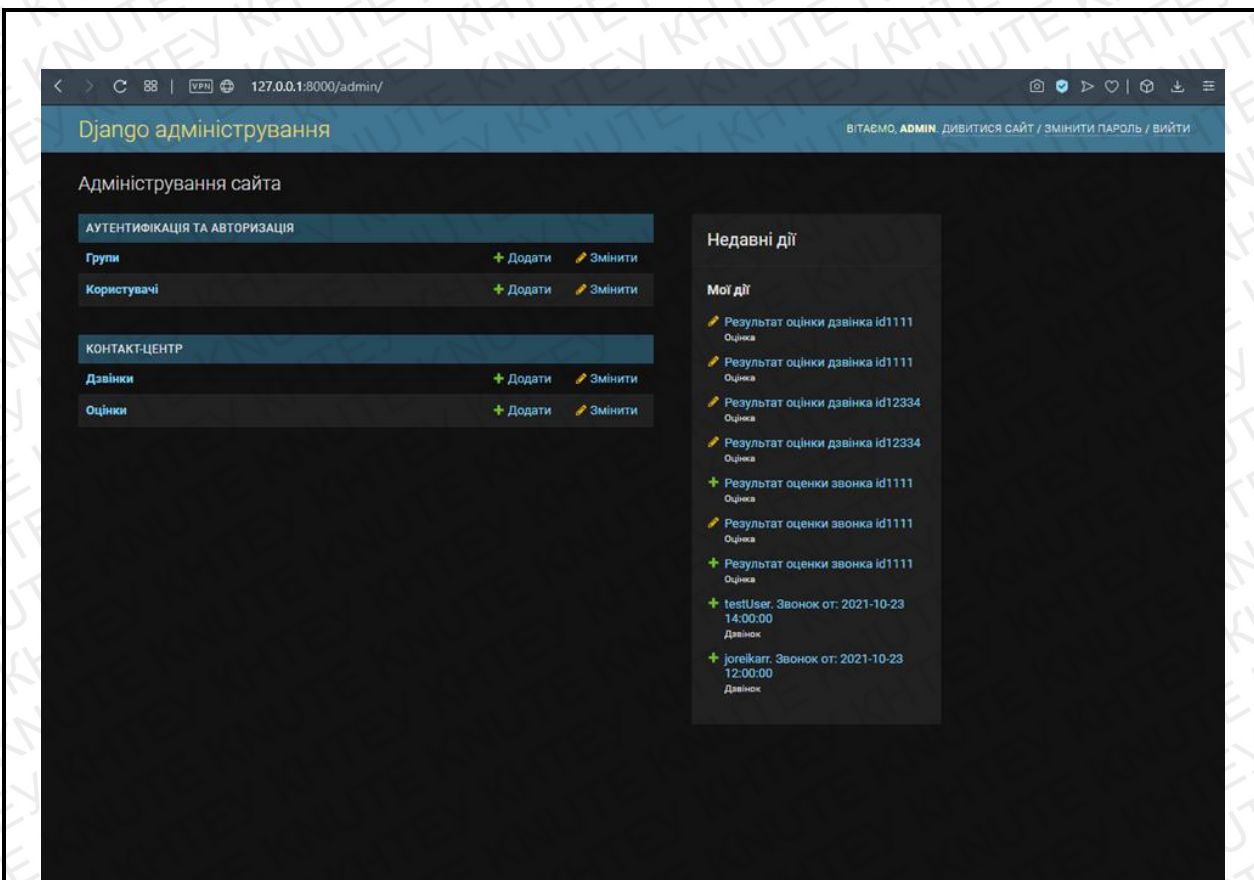


Рис. 4.7 – Сторінка панелі адміністратора

Загалом, розроблений веб-ресурс містить 9 унікальних сторінок, кожна з яких несе свій функціонал. Дизайн ресурсу мінімалістичний, представлений 4-ма основними кольорами (3 відтінки синього та сірий) та стандартним білим кольором для написів.

### 4.3. Аналіз контрольного прикладу

В якості контрольного прикладу слід розглянути процес додавання експертом нової оцінки для дзвінка. Для початку слід авторизуватися в системі. В системі заздалегідь зареєстровано експерта зі стандартною парою логіну і пароллю, admin admin.

Після авторизації слід обрати дзвінок, до якого буде створена нова оцінка.

						Аркуш
						41
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 023-03.MP	

В нашому випадку це буде дзвінок, виділений червоним квадратом на рисунку 4.8.

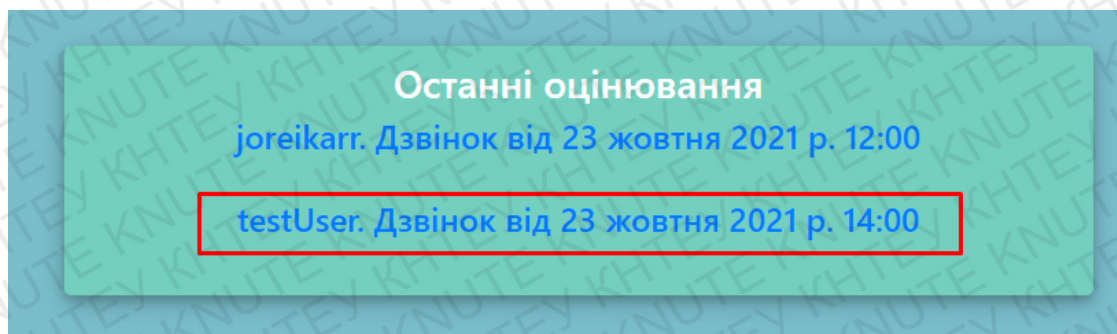


Рис. 4.8 – Вибір дзвінка для аналізу контрольного прикладу

Рисунок 4.9 показує, що в даний момент до дзвінка є лише один відгук від експерта admin.

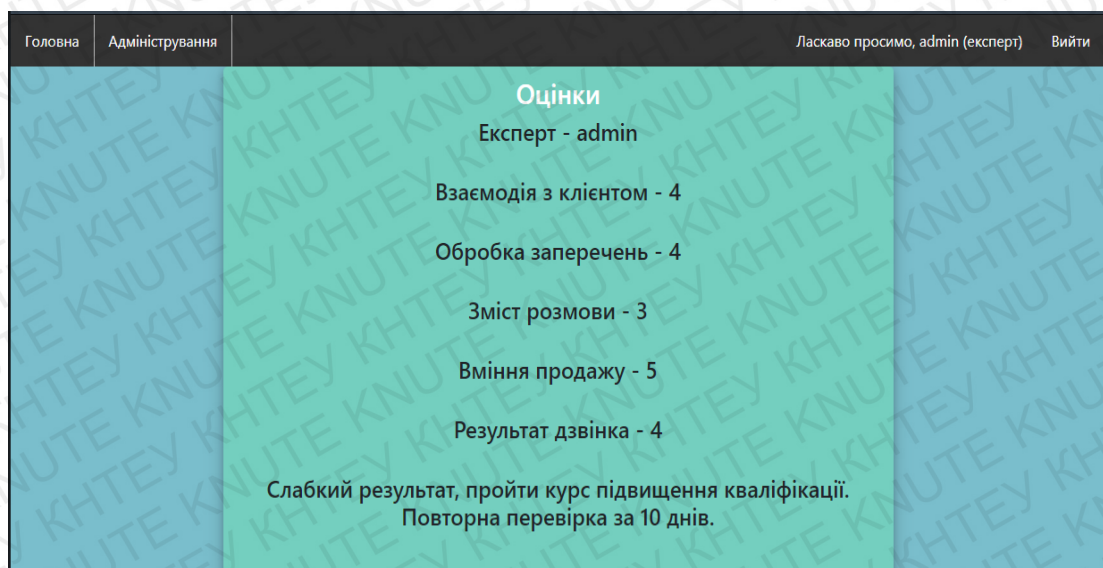


Рис. 4.9 — Оцінки обраного дзвінка

Додамо другий відгук від того ж експерта, оскільки на даний момент авторизований саме цей акаунт. Для цього натиснемо посилання внизу і заповнимо форму необхідними даними (рис. 4.10).

127.0.0.1:8000/new\_mark/id12334

### Нова оцінка

Експерт

Дзвінок

Взаємодія з клієнтом

Зміст розмови

Вміння продажу

Вміння продаж

Результат дзвінка

Оцінка и рекомендації

Помітно значне покращення якості роботи. Рекомендується підтягнути навички обробки заперечень, що надалі позитивно позначиться результативності роботи. Повторна перевірка за 14 днів.

Рис. 4.10 – Створення контрольної оцінки для обраного дзвінка

Як видно з рисунку 4.11, створений контрольний відгук було додано до списку, що означає функціональну вірність роботи веб-додатку.

						Архуш
						43
Зм.	Архуш	№ докум	Підпис	Дата	КНТЕУ 121 023-03.MP	



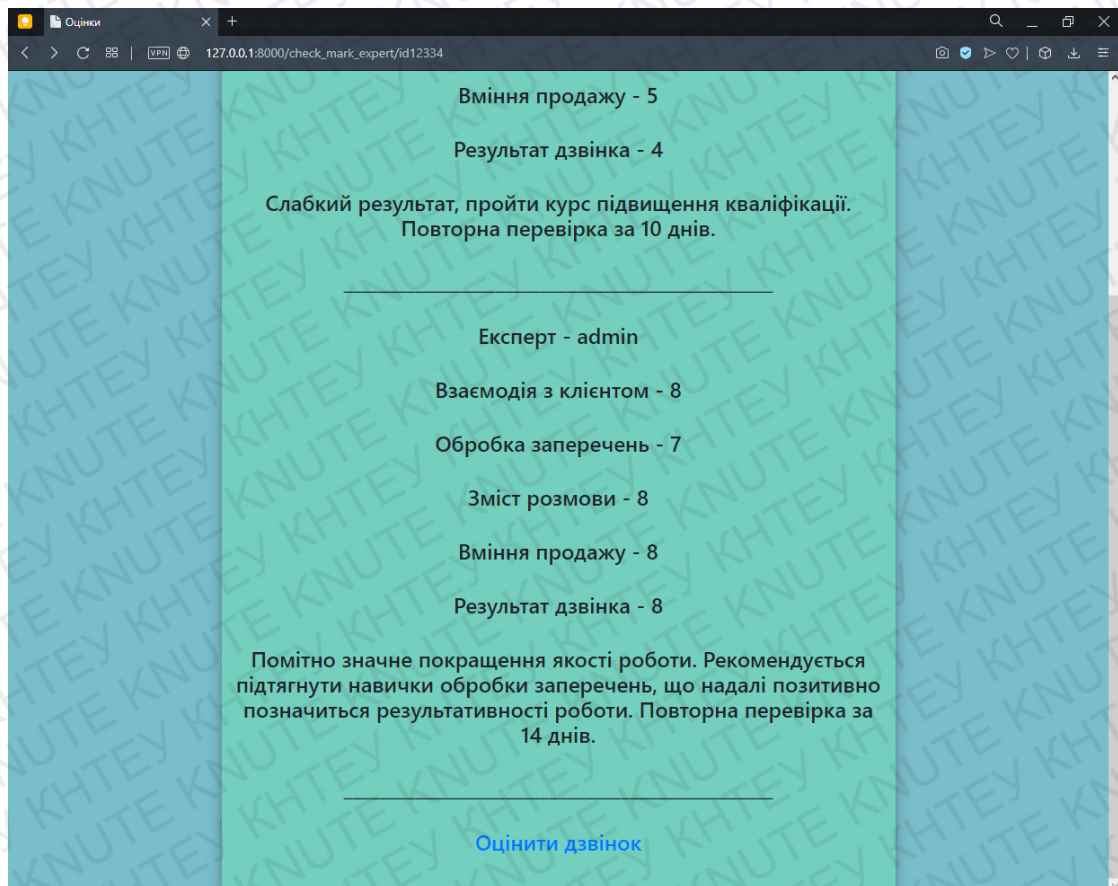


Рис. 4.11 – Контрольний відгук у списку оцінок обраного дзвінка

Отже, контрольний приклад показує повну дієздатність розробленого веб-додатку, зрозумілість графічного інтерфейсу користувача і відсутність помилок у роботі внутрішніх механізмів інформаційної системи.

					<i>КНТЕУ 121 023-03.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		44

#### 4.4. Висновок до розділу 4

Четвертий розділ роботи ілюструє процес розроблення інформаційної системи стратегічного аналізу call-центру і тестування її працездатності. Діаграма класів програмного продукту, розроблена в ході написання розділу, показує внутрішню будову системи на рівні класів та дає загальне розуміння про складність системи і взаємодії між її окремими частинами.

Окрім цього детально описано розроблений графічний інтерфейс користувача, який є основним механізмом взаємодії користувача з системою.

Проведено аналіз контрольного прикладу, який, фактично, представляє собою тестування програмного продукту методом чорного ящика.

					<i>КНТЕУ 121 023-03.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		45

## ВИСНОВКИ ТА ПРОПОЗИЦІЇ

В процесі роботи створено інформаційну систему стратегічного аналізу діяльності call-центру. Дана система покликана виконувати функціонал електронного контролю успішності роботи операторів і їх розвитку.

В процесі розроблення системи було виконано наступні завдання:

- проаналізовано аналіз контрольного прикладу, який, фактично, представляє собою тестування програмного продукту;
- розроблено веб-додаток;
- побудовано веб-додаток-аналіз варіантів використання програмного продукту, який покликаний отримати загальне розуміння про майбутній функціонал програмного забезпечення стратегічного аналізу call-центру.

За рахунок чіткого виконання завдань, поставлених на початку роботи, в результаті отримано повноцінну інформаційну систему, яка здатна виконувати закладений в неї функціонал, не має в собі неточностей роботи, протестована і повністю готова до використання в реальних умовах.

Як ідею для подальшого розвитку системи можна розглянути:

- створення механізму завантаження в систему записів розмов операторів, для підвищення зручності оцінювання дзвінків експертами;
- створення механізму побудови графічних представлень результатів оцінювань по всім операторам установи та по одному конкретному оператору, для забезпечення можливості зручного моніторингу динаміки розвитку професійних навичок і досягнень оператора.

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>КНТЕУ 121 02з-03.МР</i>			
Зав. каф.		Криворучко О.В.		01.11.21	<i>Проектування ПЗ системи стратегічного аналізу діяльності call-центру</i>	Стадія	Аркуш	Аркуші
Керівник		Рзасва С.Л.		01.11.21		ВП	46	48
Гарант		Токар В.В.		01.11.21		Факультет інформаційних технологій		
Розробив		Бойко А.А.		01.11.21		2курс, 2з група		
					<i>Висновки та пропозиції</i>			

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. "Contact Center vs Communication Center vs Call Center". EWA Bespoke Communications. 2010-03-26.
2. Science and invention in Birmingham#cite note-45
3. "The history of the call center". Call Centre Helper Magazine. 19 January 2011. Retrieved 29 November 2014.
4. Smith, Ernie (5 August 2016). "The History of the Call Center Explains How Customer Service Got So Annoying". Vice.com.
5. "The History of Call Centers Timeline". Voxjar.com. 2017-12-20.
6. Butler, David L. Bottom-Line Call Center Management. Butterworth-Heinemann.
7. Kumar, Pradeep; Schenk, Christopher Robert (2006). Paths to Union Renewal. Broadview Press. ISBN 1-55193-058-7.
8. "Improving Call Center Jobs a Top Priority for CWA Customer Service". Communication Workers of America. 2010-05-06. Retrieved 2011-02-23.
9. "Call Centre Union Busters Get Wake-Up Call". Workers Online. Retrieved 2008-07-08.
10. "Uni Global Union's call centre organising campaigns". Uni Global Union. Archived from the original on 2008-06-15. Retrieved 2008-07-08.
11. Cleveland, Brad, "Call Center Management on Fast Forward (Third Edition)", ICM Press, 2012
12. "Cactus Search - List of Call Centre Management Roles We Recruit". Cactus Search. Archived from the original on 2014-11-29.
13. Viswanathan, Ravi; Sandell, Scott (21 June 2016). "Reinventing customer service: the modern contact center". Nea.com.

					<i>КНТЕУ 121 023-03.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		01.11.21	<i>Проектування ПЗ системи стратегічного аналізу діяльності call-центру</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркуші</i>
Керівник		Рзасва С.Л.		01.11.21		<i>СВД</i>	47	48
Гарант		Токар В.В.		01.11.21		Факультет інформаційних технологій 2 курс, 23 група		
Розробив		Бойко А.А.		01.11.21				

14. "Contact Center Modernization". Genesys.com. 2016.
15. Bernier, Paula (3 August 2012). "The History and Advancement of the Contact Center and the Customer Experience". Tmcnet.com.
16. M. Popovic and V. Kovacevic (2001). "An Approach to Internet-Based Virtual Call Center Implementation". Networking — ICN 2001. Lecture Notes in Computer Science. 2093. University of Novi Sad, Yugoslavia. pp. 75–84. doi:10.1007/3-540-47728-4\_8. ISBN 978-3-540-42302-7.
17. David S. Joachim. "Computer Technology Opens a World of Work to Disabled People". New York Times. Archived from the original on 2010-03-23. Retrieved 2010-03-15.
18. Kumar, Harish (2016). "Hosted Contact / Call Center Services in Indian Telecommunications Licensing and Regulation". Researchgate: 6. doi:10.13140/RG.2.1.2931.9445.
19. "Orange currently outsources work to Indian units of Convergys Corp". The Wall Street Journal.
20. "adidas setup a dedicated customer care centre". Adidas. Archived from the original on 2013-10-27. Retrieved 2014-11-26.
21. "Audi chose Confero as an outsourced contact centre". Confero.
22. "Billing and Collections". Modern Healthcare. 2019-01-30.
23. Kasavana, Michael L.; Brooks, Richard M. (1998). Managing Front Office Operations. Educational Institute of the American Hotel & Motel Association. ISBN 9780866121798.
24. Gans, N.; Koole, G.; Mandelbaum, A. (2003). "Telephone Call Centers: Tutorial, Review, and Research Prospects" (PDF). Manufacturing & Service Operations Management. 5 (2): 79–141. doi:10.1287/msom.5.2.79.16071.

					<i>KHTEY 121 023-03.MP</i>	Арқуш
Зм.	Арқуш	№ доқум	Підпис	Дата		48

## ДОДАТКИ

### Додаток А

#### Лістинг програмного коду

```
from django.contrib import admin
from .models import marks, call
admin.site.register(marks)
admin.site.register(call)

from django.apps import AppConfig

class MainappConfig(AppConfig):
    name = 'mainApp'
    verbose_name = 'Call-центр'

from django.forms import ModelForm
from django.contrib.auth.forms import UserCreationForm
from django import forms
from django.contrib.auth.models import User
from .models import marks, call

class CreateUserForm(UserCreationForm):
    class Meta:
        model = User
        fields = ['username', 'email', 'password1', 'password2']

class CreateMark(forms.ModelForm):
    class Meta:
        model = marks
        fields = ['expert',
                 'call_FK',
                 'interaction',
                 'objections',
                 'content',
                 'skills',
                 'result',
                 'mark_value']

from django.core.validators import MaxValueValidator, MinValueValidator
from django.db import models
from django.contrib.auth.models import User
```

## Продовження дод. А

```
class call(models.Model):
    call_identifier = models.TextField('Ідентифікатор дзвінка')
    operator = models.ForeignKey(User, on_delete=models.CASCADE, verbose_name='Ідентифікатор оператора')
    call_date = models.DateField('Дата дзвонка')
    call_time = models.TimeField('Час дзвінка')
    call_duration = models.TextField('Тривалість дзвінка')

    def __str__(self):
        return self.operator.username + '. Дзвоник от: ' + str(self.call_date) + ' ' + str(self.call_time)

class Meta:
    verbose_name = 'Дзвінок'
    verbose_name_plural = 'Дзвінки'

class marks(models.Model):
    expert = models.ForeignKey(User, on_delete=models.CASCADE, verbose_name='Експерт')
    call_FK = models.ForeignKey(call, on_delete=models.CASCADE, verbose_name='Дзвінок')
    interaction = models.PositiveIntegerField(verbose_name='Взаємодія з клієнтом',
                                             validators=[MaxValueValidator(10), MinValueValidator(1)])
    objections = models.PositiveIntegerField(verbose_name='Обробка заперечень',
                                             validators=[MaxValueValidator(10), MinValueValidator(1)])
    content = models.PositiveIntegerField(verbose_name='Зміст розмови',
                                         validators=[MaxValueValidator(10), MinValueValidator(1)])
    skills = models.PositiveIntegerField(verbose_name='Вміння продажу',
                                        validators=[MaxValueValidator(10), MinValueValidator(1)])
    result = models.PositiveIntegerField(verbose_name='Результат дзвінка',
                                       validators=[MaxValueValidator(10), MinValueValidator(1)])
    mark_value = models.TextField('Оцінка')
```

## Продовження дод. А

```
def __str__(self):
    return 'Результат оцінки дзвінка ' + self.call_FK.call_identifier

class Meta:
    verbose_name = 'Оцінка'
    verbose_name_plural = 'Оцінки'
    from django.urls import path

    from . import views

    urlpatterns = [
        path("", views.index, name='index'),
        path('register/', views.regist, name='register'),
        path('cabinet/', views.operator_cabinet, name='cabinet'),
        path('new_mark/<str:callID>', views.new_mark, name='new_mark'),
        path('check_mark/<str:callID>', views.check_mark, name='check_mark'),
        path('check_mark_expert/<str:callID>', views.check_mark_expert, name='check_mark_expert'),
        path('login/', views.loginPage, name='login'),
        path('logout/', views.logoutUser, name='logout'),
    ]

    from django.shortcuts import render, redirect
    from django.contrib.auth import authenticate, login, logout
    from django.contrib import messages
    from .forms import CreateUserForm, CreateMark
    from .models import call, marks

    def loginPage(request):
        if request.method == 'POST':
            username = request.POST.get('username')
            password = request.POST.get('password')
            user = authenticate(request, username=username, password=password)

            if user is not None:
                login(request, user)
            return redirect("index")
        else:
```



## Продовження дод. А

```
messages.info(request, 'Імя користувача або пароль введені невірно')
context = {}
return render(request, 'login.html', context)

def regist(request):
    form = CreateUserForm
    if request.method == "POST":
        form = CreateUserForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('login')
    context = {'form': form}
    return render(request, 'register.html', context)

def logoutUser(request):
    logout(request)
    return redirect('login')

def index(request):
    if request.user.is_authenticated:
        call_list = call.objects.all()
        if request.user.is_superuser:
            return render(request, 'index_expert.html', {'call_list': call_list})
        else:
            return render(request, 'index.html', {'call_list': call_list})
    else:
        return redirect('login')

def check_mark(request, callID):
    cr_call = call.objects.filter(call_identifier = callID)
    marks_list = marks.objects.filter(call_FK = cr_call[0])
    context = {'marks':marks_list}
    return render(request, 'operator_check_mark.html', context)
```

## Продовження дод. А

```
def check_mark_expert(request, callID):  
    cr_call = call.objects.filter(call_identifier = callID)  
    marks_list = marks.objects.filter(call_FK = cr_call[0])  
    context = {'marks':marks_list, 'callID':callID}  
    return render(request, 'expert_check_mark.html', context)  
  
def operator_cabinet(request):  
    call_list = call.objects.filter(operator = request.user)  
    return render(request, 'operator_cabinet.html', {'call_list': call_list})  
  
def new_mark(request, callID):  
    cr_call = call.objects.filter(call_identifier = callID)  
    form = CreateMark(initial={'call_FK':cr_call[0], 'expert':request.user})
```

Продовження дод. А

```
if request.method == "POST":  
    form = CreateMark(request.POST)  
    if form.is_valid():  
        form.save()  
        return redirect('index')  
    context = {'form':form}  
    return render(request, 'new_mark.html', context)
```