

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

другого (магістерського) рівня вищої освіти

на тему: **«Розробка інформаційної системи техпідтримки клієнтів
через інтерфейс користувачів Windows Forms»**

Виконав: студент групи Іт-62

Спеціальності 126 «Інформаційні системи та
технології»

(шифр і назва)

Возняк Роман Орестович

(Прізвище та ініціали)

Керівник: к.т.н., доцент Боярчук О.В.

(Прізвище та ініціали)

Рецензент: к.т.н., доцент Бабич М.І.

(Прізвище та ініціали)

ДУБЛЯНИ-2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Другий (магістерський) рівень вищої освіти
Спеціальність 126 «Інформаційні системи та технології»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри _____

д.т.н., проф. А.М. Тригуба

«____» _____ 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу студенту

Возняка Роману Орестовичу

1. Тема роботи: «Розробка інформаційної системи техпідтримки клієнтів через інтерфейс користувачів Windows Forms»

Керівник роботи Боярчук Олег Віталійович, доцент
затверджені наказом по університету від 28.04.2023 року № 133/к-с.

2. Строк подання студентом роботи 10.01.2024 р.

3. Вихідні дані до роботи: вимоги до системи технчної підтримки клієнтів; методика проектування інформаційних систем.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити) _____

Вступ.

1. Аналіз стану техпідтримки клієнтів та наявних інформаційних систем.

2. Моделювання інформаційної системи техпідтримки клієнтів.

3. Результати проектування основних модулів інформаційної системи техпідтримки клієнтів та розробка інтерфейсу користувачів Windows Forms.

4. Охорона праці та безпека у надзвичайних ситуаціях.

5. Визначення економічної ефективності від використання інформаційної системи техпідтримки клієнтів.

Висновки та пропозиції.

Список використаної літератури.

5. Перелік ілюстраційного матеріалу (з точним зазначенням обов'язкових слайдів): аналіз стану техпідтримки клієнтів та наявних інформаційних систем; моделювання інформаційної системи техпідтримки клієнтів; результати проектування основних модулів інформаційної системи техпідтримки клієнтів та розробка інтерфейсу користувачів Windows Forms; визначення економічної ефективності.

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 5	<i>Боярчук О.В., доцент кафедри інформаційних технологій</i>		
4	<i>Городецький І.М., доцент кафедри фізики, інженерної графіки та безпеки виробництва</i>		

7. Дата видачі завдання

28 квітня 2023 р.

Календарний план

№ з/п	Назва етапів кваліфікаційної роботи	Терміни виконання етапів роботи	Примітка
1	<i>Написання першого розділу</i>	<i>28.04-20.05.23</i>	
2	<i>Виконання другого розділу та аркушів ілюстраційного матеріалу до нього</i>	<i>21.05-14.08.23</i>	
3.	<i>Виконання третього розділу та аркушів ілюстраційного матеріалу до нього</i>	<i>15.08.23-10.07.24</i>	
4.	<i>Написання розділу «Охорона праці та безпека у надзвичайних ситуаціях»</i>	<i>11.07-31.08.24</i>	
5.	<i>Оцінення ефективності запропонованої системи</i>	<i>01.09-31.10.24</i>	
6.	<i>Завершення оформлення розрахунково-пояснювальної записки та аркушів ілюстраційного матеріалу</i>	<i>01-30.11.24</i>	
7.	<i>Завершення роботи в цілому</i>	<i>01-10.12.24</i>	

Студент _____ Возняк Р.О.
(підпис)

Керівник роботи _____ Боярчук О.В.
(підпис)

УДК 004.41:004.927:004.89

Розробка інформаційної системи техпідтримки клієнтів через інтерфейс користувачів Windows Forms.

Возняк Р.О. Кафедра інформаційних технологій – Дубляни, ЛНУП, 2024.

Кваліфікаційна робота: 63 с. текст. част., 11 рис., 5 табл., 15 арк. ілюстраційного матеріалу, 30 джерел.

У кваліфікаційній роботі проведено комплексний аналіз сучасного стану технічної підтримки клієнтів, виявлено основні проблеми та недоліки існуючих систем і інструментів. На основі проведеного аналізу визначено завдання, які необхідно вирішити для підвищення ефективності роботи служб техпідтримки. Для цього розроблено інформаційну систему технічної підтримки клієнтів, яка базується на концептуальній і логічній моделях, що відображають структуру і взаємозв'язки основних компонентів. Розробка моделі системи включала проектування діаграм варіантів використання та бізнес-процесів у нотаціях IDEF0 і IDEF3.

У роботі також виконано проектування основних модулів системи, таких як модулі прийому заявок, обробки заявок, управління користувачами та категоріями заявок. Для реалізації графічного інтерфейсу користувачів використано платформу Windows Forms, що забезпечує високу функціональність, зручність використання та адаптивність системи до потреб організацій. Створений інтерфейс дозволяє користувачам ефективно взаємодіяти із системою, оперативно обробляти заявки клієнтів і керувати всіма аспектами технічної підтримки.

Ключові слова: технічна підтримка клієнтів, інформаційна система, моделювання, інтерфейс користувачів, Windows Forms.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ СТАНУ ТЕХПІДТРИМКИ КЛІЄНТІВ ТА НАЯВНИХ ІНФОРМАЦІЙНИХ СИСТЕМ	9
1.1. Аналіз існуючого стану техпідтримки клієнтів організацій	9
1.2. Аналіз існуючих інструментів технічної підтримки клієнтів	10
1.3. Основні проблеми існуючих систем технічної підтримки клієнтів	14
1.4. Завдання кваліфікаційної роботи	15
РОЗДІЛ 2. МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТЕХПІДТРИМКИ КЛІЄНТІВ.....	17
2.1. Розробка концептуальної моделі системи техпідтримки клієнтів	17
2.2. Розробка логічної моделі системи техпідтримки клієнтів.....	18
2.3. Розробка діаграми варіантів використання	20
2.4. Розробка діаграм нотацій моделювання IDEF0 та IDEF3.....	22
РОЗДІЛ 3. РЕЗУЛЬТАТИ ПРОЄКТУВАННЯ ОСНОВНИХ МОДУЛІВ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТЕХПІДТРИМКИ КЛІЄНТІВ ТА РОЗРОБКА ІНТЕРФЕЙСУ КОРИСТУВАЧІВ WINDOWS FORMS	27
3.1. Доцільність проєктування основних модулів інформаційної системи техпідтримки клієнтів та переваги інтерфейсу користувачів Windows Forms	27
3.2 Вибір засобів для проєктування основних модулів інформаційної системи техпідтримки клієнтів та розробки інтерфейсу користувачів Windows Forms	29
3.3. Проєктування основних модулів інформаційної системи	32
3.3.1. Модуль прийому заявок	32
3.3.2. Модуль обробки заявок	34
3.3.3. Модуль управління користувачами.....	36
3.3.4. Модуль управління категоріями заявок.....	39
3.4. Розробка інтерфейсу користувачів Windows Forms	41
РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ	45

4.1. Аналіз небезпек та шкідливих виробничих під час розробки інформаційної системи	45
4.2. Розробка заходів щодо покращення умов праці	46
4.3. Розробка логічно-імітаційної моделі процесу виникнення травм під час монтажу інтелектуальної інформаційної системи	47
4.4. Розробка заходів щодо безпеки у надзвичайних ситуаціях	50
РОЗДІЛ 5. ВИЗНАЧЕННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД	
ВИКОРИСТАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТЕХПІДТРИМКИ	
КЛІЄНТІВ	
ВИСНОВКИ І ПРОПОЗИЦІЇ	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	54
ДОДАТКИ	58
Додаток А. Фрагмент коду для інтерфейсу клієнта	61
	62

ВСТУП

Розвиток інформаційних технологій дозволяє значно підвищити ефективність управління бізнес-процесами та покращити якість обслуговування клієнтів. Особливо актуальним це для технічної підтримки, де оперативність і зручність взаємодії між клієнтами та технічними спеціалістами є визначальними факторами [1]. Зростання кількості клієнтів, підвищення продуктивності обладнання, а також потреби обробки великих обсягів запитів вимагають створення автоматизованих інформаційних систем для виконання цих завдань.

Інформаційні системи техпідтримки забезпечують швидкий і зручний спосіб прийому, обробку та моніторинг звернень клієнтів, а також управління статусом запитів. Завдяки таким системам компанія може значно підвищити рівень задоволеності клієнтів, мінімізувати людський фактор у процесі обслуговування та оптимізувати використання ресурсів.

Розробка інформаційної системи з використанням Windows Forms для створення інтерфейсу користувача є доцільною з кількох причин. По-перше, Windows Forms є перевіреною та надійною технологією для створення настільних додатків у середовищі Windows [8]. Вона забезпечує гнучкість у розробці та дозволяє створювати зручний графічний інтерфейс із різноманітними інтерактивними елементами, такими як кнопки, списки, текстові поля тощо. Крім того, використання цієї технології дозволяє скоротити витрати на розробку, крім того, вона має низькі вимоги до розробників та багатий набір готових інструментів.

Актуальність теми роботи зумовлена зростаючою потребою в існуючому процесі технічної підтримки. Ручне управління заявками та комунікаціями із клієнтами часто призводить до помилок, затримок і незадоволеності користувачів. Впровадження автоматизованої інформаційної системи вирішує ці проблеми, забезпечуючи централізоване управління запитами, швидкий доступ до історії обслуговування клієнтів, а також підвищуючи прозорість і підконтрольність процесів.

Метою даної кваліфікаційної роботи є розробка інформаційної системи техпідтримки клієнтів через інтерфейс Windows Forms, яка забезпечує створення, обробку та моніторинг клієнтських заявок, а також підтримує взаємодію між клієнтами та технічними спеціалістами.

Об'єктом дослідження є процеси управління заявками на технічну підтримку в організації. Предметом дослідження є методи та технології автоматизації цих процесів.

Практичне значення розробленої інформаційної системи дозволяє уможливити її впровадження в будь-яку організацію, що надає технічну підтримку, для підвищення швидкості та якості обслуговування клієнтів.

Таким чином, створення інформаційної системи техпідтримки клієнтів є вагомим етапом у розвитку сучасних організацій, які прагнуть покращити комунікацію з клієнтами, мінімізувати час вирішення технічних проблем і підвищити загальну ефективність своєї роботи.

РОЗДІЛ 1.

АНАЛІЗ СТАНУ ТЕХПІДТРИМКИ КЛІЄНТІВ ТА НАЯВНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

1.1. Аналіз існуючого стану техпідтримки клієнтів організацій

Технічна підтримка клієнтів є невід’ємною складовою сучасного бізнесу, особливо у сферах, пов’язаних із технологіями, послугами та сервісами. Забезпечення якісної та оперативної технічної підтримки підтримує рівень задоволеності клієнтів, їхню лояльність, а також на репутацію компанії.

Проаналізовано сучасний стан організації технічної підтримки клієнтів, зокрема використаних традиційних та автоматизованих підходів, інструментів, а також проблем та викликів, які використовуються в процесі обслуговування клієнтів.

На основі аналізу компаній, які активно працюють у сфері технічної підтримки, можна виділити три основні підходи, які вони використовують (рис. 1.1).

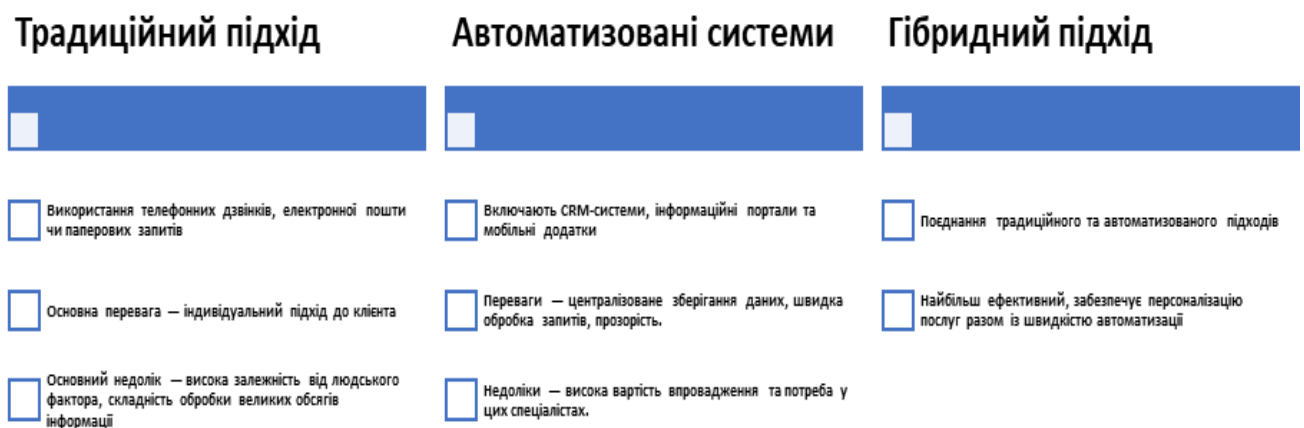


Рисунок 1.1 – Основні підходи у сфері технічної підтримки

Традиційний підхід передбачає використання телефонних дзвінків, електронної пошти чи паперових запитів. Основна його перевага у тому, що забезпечує індивідуальний підхід до клієнта. Основний недолік полягає в тому,

що наявна висока залежність від людського чинника, складність обробки великих обсягів інформації.

Заслуговують на увагу автоматизовані системи технічної підтримки клієнтів організацій. Вони включають CRM-системи, інформаційні портали та мобільні додатки. Основними перевагами таких систем є централізоване зберігання даних, швидка обробка запитів, прозорість. До недоліків належить висока вартість впровадження та потреба у окремих спеціалістах.

Гібридний підхід передбачає поєднання традиційного та автоматизованого підходів. Він є найбільш ефективний, забезпечує персоналізацію послуг разом із швидкістю автоматизації.

1.2. Аналіз існуючих інструментів технічної підтримки клієнтів

Нами виконано аналіз існуючих інструментів технічної підтримки, які представлено у табл. 1.1.

Таблиця 1.1 – Існуючі інструменти для технічної підтримки клієнтів

Назва інструменту	Функціонал	Переваги	Недоліки
Zendesk [26]	CRM-система для управління заявками	Інтеграція з іншими каналами комунікації	Висока вартість підписки
Freshdesk [24]	Портал підтримується автоматизацією	Простота використання, гнучкість	Обмежений функціонал у безкоштовній версії
Робочий стіл Zoho [30]	Управління запитами та аналітика	Можливість інтеграції з Zoho CRM	Складний інтерфейс для початківців
Microsoft Dynamics 365 [27]	Професійна платформа підтримки	Масштабованість, аналітика	Складність налаштування

На рисунку 1.2 показано типовий інтерфейс системи Zendesk, який демонструє функціональне управління запитами, включаючи статус перегляду, пріоритету та коментарів до заявок [26].

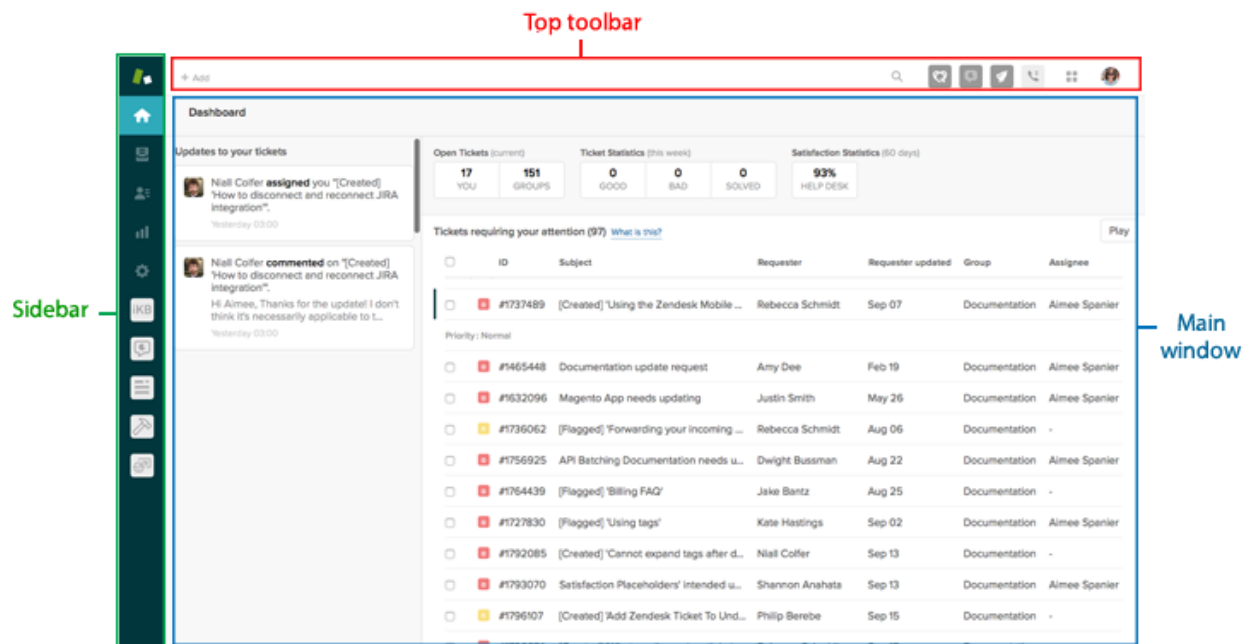


Рисунок 1.2 – Типовий інтерфейс системи Zendesk [26]

Коли відкриваєте службу підтримки Zendesk, з'являється інформаційна панель агента. Інтерфейс має такі розділи:

- Бічна панель, область внизу лівої частини інтерфейсу, з кнопками навігації до основних функцій;
- Верхня панель інструментів, область у верхній частині інтерфейсу, з вкладками для відкриття та переходу між заявками, а також швидкими посиланнями для пошуку та інших програм;
- Головне вікно, де відображається більшість вашого вмісту.

СТІ – це структура, яка дозволяє створювати та інтегрувати телефонну систему з Freshdesk [24]. Щоб скористатися функціями системи телефонії, доведеться створити спеціальний додаток. Потім можна відобразити цю спеціальну програму всередині iframe і дозволити своїм агентам приймати дзвінки з будь-якої сторінки Freshdesk.

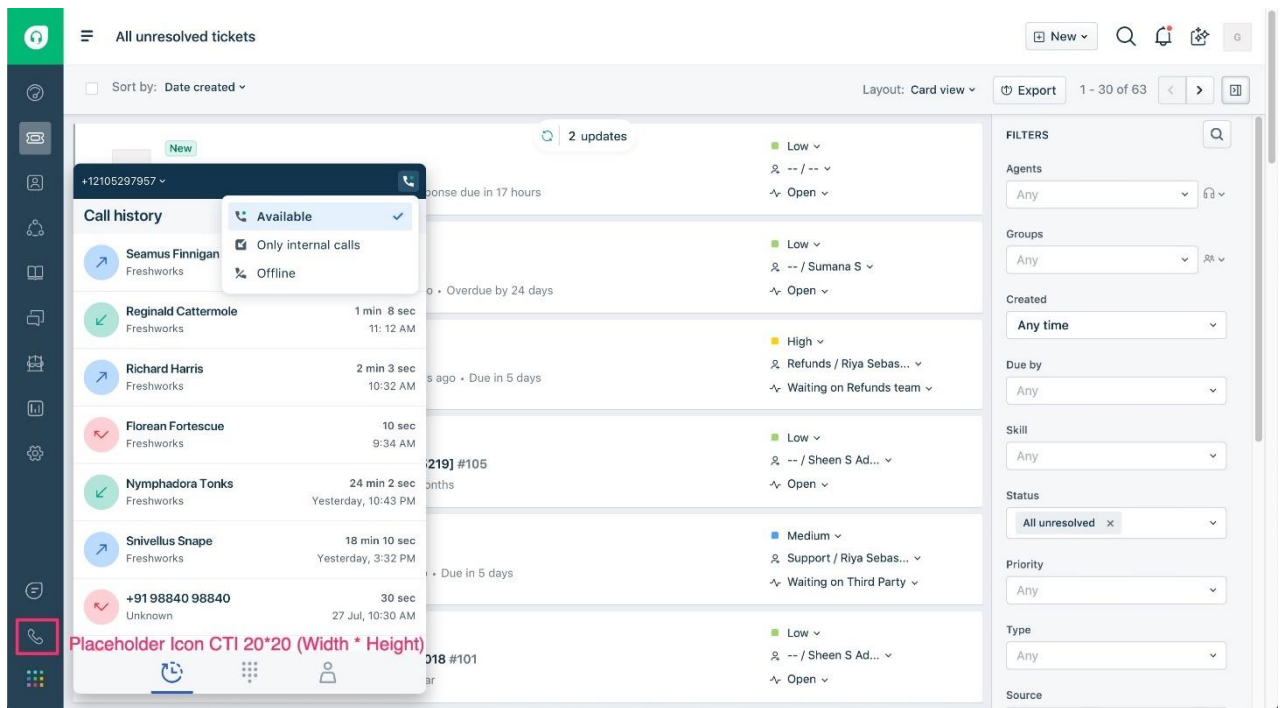


Рисунок 1.3 – Типовий інтерфейс системи Freshdesk [24]

Клієнт (або його розробник) може звернутися до цієї сторінки, щоб швидко розпочати створення програм і використати доступний заповнювач, щоб відобразити його всередині iframe. доведеться обробляти атрибути програми в самому маніфесті програми, оскільки в СТІ немає окремої сторінки адміністратора. Щоб отримати квитки, контакти чи будь-які контекстні дані, які агенти знадобляться в програмі, можна скористатися Freshdesk API . Висоту та ширину (максимум 500 пікселів) iframe можна налаштувати. Стандартне значення – 400 пікселів.

Freshcaller не можна ввімкнути, коли ввімкнено програму СТІ. Після обробки автентифікації можна створити свою програму так, як хочете, щоб вона функціонувала, і використовувати розташування для рендерингу програми.

Zoho CRM Plus – це інтегрована платформа для взаємодії з клієнтами, яка дозволяє створити високоякісний досвід на кожному етапі співпраці з клієнтом, сприяючи формуванню їхньої лояльності (рис. 1.4). Вона об'єднує команди, відповідальні для маркетингу, продажу та технічної підтримки, забезпечуючи їх усіма необхідними інструментами для досягнення максимальної ефективності.

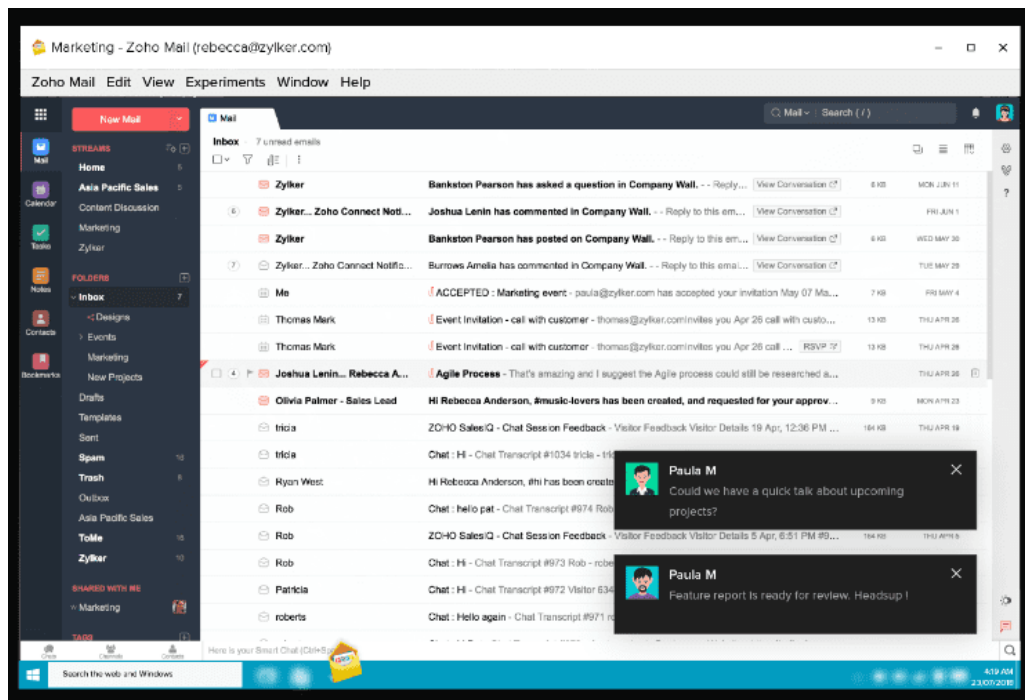


Рисунок 1.4 – Типовий інтерфейс Zoho Mail Desktop Lite [30]

Zoho Mail Desktop Lite – це безкоштовна програма для настільних комп'ютерів, яка дозволяє отримувати доступ до Zoho Mail одночасно з ПК чи ноутбука. Вона поєднує звичний інтерфейс веб-версії з покращеною швидкістю та зручністю. Завдяки синхронізації з веб-додатком користувачі можуть безперешкодно перемикатися між платформами. Вона має легкий доступ до поштової скриньки всього одним кліком. Миттєві десктопні повідомлення та значки повідомлень у системному для моніторингу особистих і робочих листів. Мінімалістичний дизайн без зайвих рамок, який дозволяє зосередитися на роботі з електронною поштою, не відволікаючись у браузері. Завдяки локальному розміщенню програми час завантаження зводиться до мінімуму, що забезпечує швидкий та ефективний досвід використання.

Microsoft Dynamics 365 – це універсальне програмне рішення, яке дозволяє автоматизувати ключові бізнес-процеси, такі як фінансове управління, управління проектами, продажами та маркетингом [27]. Компанія TechExpert спеціалізується на впровадженні та адаптації Dynamics 365 для забезпечення максимальної ефективності роботи клієнтів. Підходять до кожного об'єкта індивідуально, гарантуючи високу якість виконання та термінів.

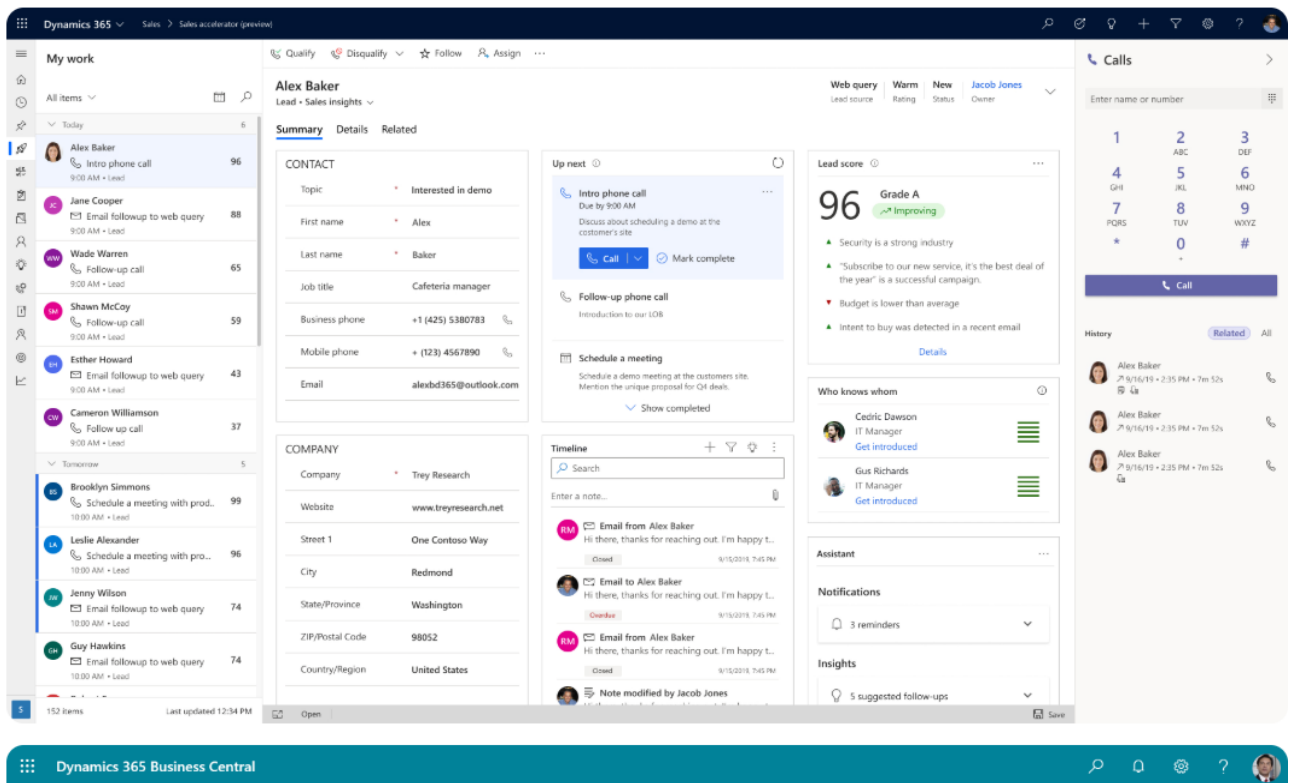


Рисунок 1.5 – Типовий інтерфейс Microsoft Dynamics 365 [27]

TechExpert пропонує повний спектр послуг з налаштування, інтеграції та доопрацювання рішень на базі Microsoft Dynamics 365. Спеціалісти мають досвід роботи зі спільними галузями та бізнес-процесами, що дозволяє адаптувати систему під конкретні потреби замовника.

1.3. Основні проблеми існуючих систем технічної підтримки клієнтів

До основних проблем існуючих систем технічної підтримки клієнтів належать:

- висока вартість впровадження – професійні CRM-системи, як-от Zendesk чи Microsoft Dynamics, потребують значних інвестицій на етапі впровадження та навчання персоналу;
- складність використання – багато систем мають складний інтерфейс, що потребує додаткового навчання.

➤ обмежена функціональність у безкоштовних версіях – наприклад, Freshdesk у безкоштовній версії має обмеження щодо кількості користувачів та функцій автоматизації.

➤ проблеми персоналізації – у масових автоматизованих системах часто втрачається індивідуальний підхід до клієнта.

Аналіз існуючого стану технічної підтримки клієнтів організацій показав, що впровадження автоматизованої системи є бажаним етапом підвищення ефективності обслуговування клієнтів. Популярні інструменти, такі як Zendesk, Freshdesk і Zoho Desk, володіють широким функціоналом, але їхня вартість і надійність налаштування залишаються значним бар'єром для більшості організацій. Використання Windows Forms для створення інформаційної системи техпідтримки дозволяє запропонувати альтернативні рішення, орієнтовані на локальні потреби організацій, із меншими витратами на впровадження та можливість гнучкої адаптації.

1.4. Завдання кваліфікаційної роботи

Метою даної кваліфікаційної роботи є розробка інформаційної системи технічної підтримки клієнтів, яка забезпечує ефективну обробку запитів клієнтів та управління процесами технічної підтримки. Система буде реалізована з використанням інтерфейсу Windows Forms, що дозволить створити зручний настільний додаток для операторів техпідтримки.

Для досягнення поставленої мети обов'язковим є такі основні завдання:

1. Аналіз існуючих систем технічної підтримки. Провести огляд сучасних інструментів, методів і технологій, які використовують в системах техпідтримки. Виявити переваги та недоліки популярних рішень, таких як Microsoft Dynamics 365, Zendesk, Freshdesk та інших. Обґрунтувати вибір Windows Forms як платформи для реалізації інформаційної системи.

2. Проектування інформаційної системи. Виконати проектування основних модулів системи – прийом заявок, моніторинг статусу запитів, управління заявками та звітність. Описати взаємодію між користувачем (оператором техпідтримки) і системою. Забезпечити можливість гнучкого налаштування для адаптації під різні бізнес-процеси.

3. Реалізація функціональних модулів системи. Розробити інтерфейс для створення, обробки та перегляду заявок. Реалізувати зміну функції статусу заявки (наприклад, «Відкрита», «В обробці», «Закрита»). Забезпечити зручний механізм пошуку та фільтрації заявок.

4. Проведення тестування інформаційної системи. Перевірити працездатність системи на різних сценаріях роботи, включаючи створення та обробку заявок. Визначити недоліки в логіці або дизайні системи та усунути їх.

5. Розробити заходи щодо охорони праці та безпеки у надзвичайних ситуаціях.

6. Визначити економічну ефективність від розробок.

Виконана робота забезпечує значне підвищення ефективності та зручності процесів технічної підтримки клієнтів, скорочуючи час обробки запитів та знижуючи навантаження на персонал. Розроблена система сприяє підвищенню задоволеності клієнтів, забезпечуючи стабільну та надійну роботу в реальних умовах експлуатації.

РОЗДІЛ 2. МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТЕХПІДТРИМКИ КЛІЄНТІВ

2.1. Розробка концептуальної моделі системи техпідтримки клієнтів

Концептуальне моделювання – це процес створення абстрактної моделі, яка описує інформаційний зміст системи, незалежно від її фізичної реалізації. Такий підхід забезпечує формування семантичної моделі, що відображає основні елементи та зв'язки між ними в контексті аналізованої тематики. У нашому випадку йдеться про систему технічної підтримки користувачів. Концептуальне моделювання дозволяє структурувати дані та встановлювати взаємозв'язки між об'єктами баз даних, які відображають функціональність системи.

У таблицях 2.1 і 2.2 представлено характеристику бази даних основних об'єктів та їх атрибутів, які використані у розробці системи технічної підтримки.

Таблиця 2.1 – Характеристика об'єктів бази даних

Назва об'єкта	Характеристика об'єкта
Користувач (user)	Зберігає інформацію про ім'я користувача, його контактні дані та роль у системі.
Заявка (ticket)	Містити інформацію про створені клієнтами заявки, включаючи їхній статус, дату створення тощо.
Процеси (process)	Зберігає дані про процеси, активні на комп'ютері користувача на момент створення заявки.
Тип (type)	Містити інформацію про категорії або типи заявок, наприклад, звичайна чи термінова.

Таблиця 2.2 – Опис атрибутів об'єктів

Назва	Характеристика
-------	----------------

атрибута	атрибута
user_id	Первинний ключ об'єкта user, який ідентифікує користувача.
username	Ім'я користувача, який створив заявку.
email	Електронна адреса користувача для зворотного зв'язку.
ticket_id	Первинний ключ об'єкта ticket, що ідентифікує заявку.
headline	Заголовок заявки, що коротко описує проблему.
description	Детальний опис проблеми, зазначений користувачем.
status	Статус заявки, наприклад, «Відкрита», «Закрита».
creation_date	Дата та час створення заявки.
type_id	Зв'язок із таблицею type, який позначає категорію заявки.
process_id	Первинний ключ об'єкта process, що ідентифікує активні процеси на момент створення заявки.
process_name	Назва процесу, активного на комп'ютері користувача.
PID	Ідентифікатор процесу в операційній системі.
memory_usage	Обсяг пам'яті, який використовує процес.
type_description	Тип заявки: звичайна чи термінова.

Розглянемо зв'язки між об'єктами бази даних. Користувач (user) створює заявку (ticket). Заявка (ticket) може включати кілька процесів (process), що були активні на комп'ютері користувача під час створення заявки. Заявка (ticket) пов'язана з категорією або типом (type). Ці зв'язки забезпечують можливість комплексного управління заявками та збору необхідної інформації для вирішення проблем користувачів.

2.2. Розробка логічної моделі системи техпідтримки клієнтів

Логічне моделювання є важливим етапом у розробці інформаційних систем. Воно передбачає створення моделі даних, яка є незалежною від

конкретного типу системи керування базами даних (СКБД) та фізичних аспектів реалізації. Ця модель дозволяє організувати дані таким чином, щоб вони ефективно підтримували бізнес-процеси.

На цьому етапі логічна модель даних створюється шляхом уточнення та трансформації концептуальної моделі, розробленої на попередньому етапі. Основна увага приділяється визначенню зв'язків між об'єктами та атрибутами, які забезпечують коректне зберігання та обробку даних у системі.

У розробці інформаційної системи технічної підтримки клієнтів через інтерфейс Windows Forms логічна модель бази даних складається з наступних основних об'єктів:

- Користувач (user) – зберігає інформацію про користувачів системи;
- Заявка (ticket) – містить дані про заявки, створені клієнтами;
- Процеси (process) – відображає дані про активні процеси на комп'ютері користувача під час створення заявки;
- Тип (type) – містить інформацію про категорії або типи заявок.

Проводимо визначення первинних та зовнішніх ключів:

- Користувач (user): user_id – первинний ключ;
- Заявка (ticket): ticket_id – первинний ключ; user_id, type_id – зовнішні ключі.
- Процеси (process) – process_id – первинний ключ; ticket_id – зовнішній ключ.
- Тип (type): type_id – первинний ключ.

Результат логічного моделювання представлений на рисунку 2.1. Він демонструє взаємозв'язки між основними об'єктами бази даних, а також використання первинних і зовнішніх ключів для забезпечення цілісності даних.

На рисунку 2.1 представлено логічну модель бази даних, що відображає взаємозв'язки між об'єктами user, ticket, process і type. Ця модель забезпечує структуровану організацію даних, необхідних для ефективно роботи системи технічної підтримки.

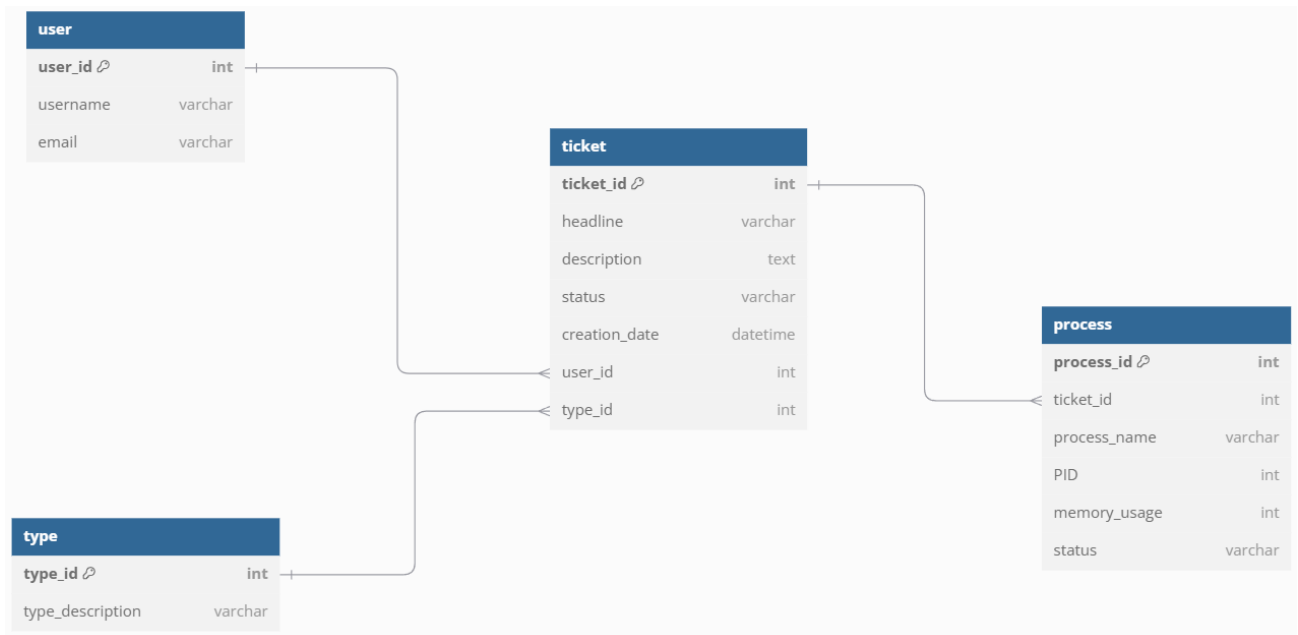


Рисунок 2.1 – Логічна модель бази даних

Розробка логічної моделі бази даних є важливим етапом у створенні інформаційної системи технічної підтримки клієнтів. Вона дозволяє структуровано представити моделі даних, забезпечуючи інтеграцію ключових об'єктів та їх атрибутів. Модель спрощує реалізацію системи та її подальше використання.

2.3. Розробка діаграми варіантів використання

Діаграма варіантів використання (use case diagram) є важливим інструментом для моделювання функціональних можливостей системи та взаємодії користувачів з нею. Вона дозволяє наочно продемонструвати ключові функції, які виконуються у системі технічної підтримки клієнтів, та визначити ролі користувачів.

Розробка діаграми варіантів використання є ключовим етапом в аналізі функціональних вимог до системи. Вона дозволяє візуалізувати сценарії взаємодії користувачів із системою, визначити ролі користувачів (акторів) та описати функції (варіанти використання), які забезпечує система. У випадку

розробки інформаційної системи технічної підтримки клієнтів через інтерфейс Windows Forms, діаграма варіантів використання допомагає описати основні функціональні можливості системи та її взаємодію з користувачами.

У розробленій системі технічної підтримки визначено три основні ролі (актори):

1) Клієнт – користувач, який створює заявки на технічну підтримку. Ця роль охоплює дії, пов'язані з описом проблеми та поданням заявки через зручний інтерфейс.

2) Інженер технічної підтримки – фахівець, який обробляє створені заявки. Його основні завдання включають аналіз поданих заявок, фіксацію рішень та зміну статусу заявки (наприклад, «Відкрита», «Закрита»).

3) Адміністратор – користувач із розширеними правами, який управляє системними даними. Його функції включають додавання, редагування та видалення користувачів, а також управління категоріями заявок (типами).

На основі функціональних вимог до системи було визначено такі ключові варіанти використання:

1) Створення заявки – клієнт створює заявку, вводячи інформацію про проблему. Цей процес включає введення заголовка, опису проблеми, а також прикріплення додаткових матеріалів, таких як скріншоти.

2) Обробка заявки – інженер технічної підтримки переглядає деталі заявки, аналізує її, фіксує рішення та змінює статус заявки. Наприклад, заявка може бути переведена зі статусу «Відкрита» в «Закрита».

3) Управління користувачами – адміністратор додає нових користувачів до системи, редагує їхні дані або видаляє облікові записи за потреби.

4) Управління категоріями заявок – Адміністратор визначає категорії або типи заявок (наприклад, «Термінова» чи «Планова») для подальшої класифікації та зручності обробки.

Кожен із варіантів використання відповідає певній взаємодії між актором і системою. Наприклад, клієнт створює заявку, яка потім передається інженеру

технічної підтримки. Інженер працює з цією заявкою, змінюючи її статус та фіксуючи рішення. Адміністратор виконує дії, які забезпечують правильне функціонування системи, додаючи нових користувачів і типи заявок.

Діаграма варіантів використання для системи технічної підтримки клієнтів зображена на рисунку 2.2. Вона містить основні актори, варіанти використання та взаємозв'язки між ними.



Рисунок 2.2 – Діаграма варіантів використання інформаційної системи технічної підтримки

Діаграма варіантів використання дозволяє зрозуміти ключові функції системи, оптимізувати її дизайн та забезпечити відповідність функціональних можливостей реальним потребам користувачів. Вона також спрощує комунікацію між розробниками та зацікавленими сторонами, адже наочно демонструє, як саме користувачі взаємодіють із системою.

Детальний аналіз варіантів використання та їх графічне представлення є основою для подальшої розробки системи, зокрема для реалізації інтерфейсу та моделювання бази даних. Це забезпечує ефективне планування, розробку та тестування всіх компонентів інформаційної системи технічної підтримки клієнтів.

2.4. Розробка діаграм нотацій моделювання IDEF0 та IDEF3

Моделювання процесів у розробці інформаційних систем є важливим інструментом для формалізації і вдосконалення бізнес-процесів. Для опису функціональних аспектів системи використовуються нотації IDEF0 та IDEF3, які

дозволяють розробити структуровану модель системи з акцентом на її функції та взаємозв'язки між ними.

Нотація IDEF0 (Integration Definition for Function Modeling) використовується для опису функціональної моделі системи. Основна мета цієї нотації полягає у графічному представленні процесів, які виконуються в системі, а також ресурсів, що використовуються для їх виконання.

Для інформаційної системи технічної підтримки клієнтів діаграма IDEF0 ілюструє основні функції, такі як обробка заявок, управління користувачами та управління категоріями заявок. У моделі IDEF0 кожен процес представлений у вигляді блока, який має чотири основні компоненти:

Виконаємо деталізацію складових для діаграм IDEF0:

1. Вхідні дані – це інформація або матеріали, необхідні для запуску та виконання певної функції. Вони є основою для виконання завдань і мають надходити із зовнішнього середовища або з іншими функціями в системі.

Для інформаційної системи технічної підтримки клієнтів:

- дані від клієнта (заголовок заявки, опис проблеми, прикріплені файли (наприклад, скріншоти чи журнали));
- дані про поточний стан системи (лог файли, стан комп'ютера клієнта (завантаження CPU, RAM));
- інформація про тип заявки (класифікація (звичайна/термінова)).

2. Вихідні дані – це результат виконання функцій, які є необхідними для подальших етапів роботи системи або передачі в зовнішнє середовище.

Для системи технічної підтримки клієнтів вихідні дані є:

- Для клієнта – підтвердження створення заявки з унікальним ідентифікатором, відповідь від інженера технічної підтримки, інформація про статус заявки (наприклад, «Закрита», «Очікує відповіді»).
- Для адміністратора – звіт про активність інженерів, аналітичні дані про кількість заявок за категоріями.
- Для інженера – детальна інформація про заявки та рекомендації щодо рішення проблеми (на основі знань).

3. Механізми – це ресурси, які забезпечують виконання функцій. Вони можуть бути як технічними (програмне забезпечення, обладнання), так і людськими (персональними, командами).

Для інформаційної системи технічної підтримки клієнтів:

➤ Технічні механізми – програмне забезпечення Windows Forms, база даних для зберігання заявок та процесів, сервери, які забезпечують виконання запитів, інструменти для моніторингу системного стану комп'ютерів.

➤ Людські механізми – клієнти, які створюють заявки, інженери, що обробляють заявки, адміністратори, які керують системою.

4. Управління вироблення правил, стандартів та процедур, які забезпечують коректне виконання функцій. Ці правила можуть бути як системними, так і організаційними.

Для інформаційної системи технічної підтримки клієнтів:

➤ Політики та процедури – правила прийняття заявок (формат, обов'язкові поля), регламент обробки термінових заявок (наприклад, час відповіді не більше 1 року).

➤ Системні обмеження – перевірка вхідних даних на коректність (відсутність порожніх полів), ролі користувачів (доступ до певних функцій відсутній для інженерів чи адміністраторів).

➤ Регламенти безпеки – шифрування даних при передачі через мережу, авторизація користувачів в системі.

На рисунку 2.3 представлено діаграму IDEF0, що демонструє основні функції системи технічної підтримки клієнтів.

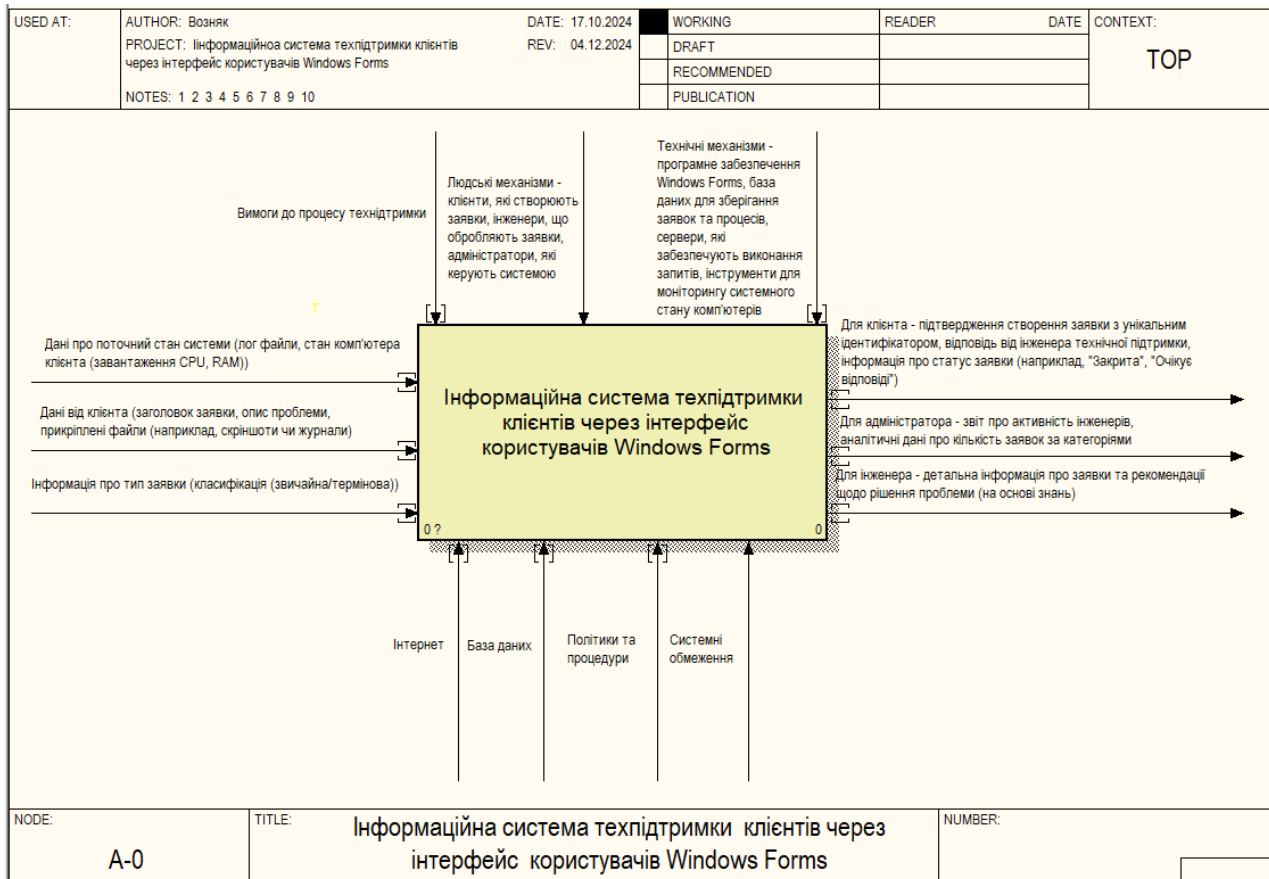


Рисунок 2.3 – Діаграма IDEF0 для системи технічної підтримки клієнтів

На діаграмі IDEF0 перший рівень декомпозиції демонструє основну функцію системи «Управління заявками технічної підтримки» та чотири підфункції: прийом заявки, обробка заявки, управління користувачами, управління категоріями заявок. Цей рівень дає змогу зрозуміти структуру функціонування системи та взаємодію між ключовими процесами.

Ця деталізація чітко підтримує основні процеси, їх входи, виходи, механізми виконання та елементи управління. Це є основою для подальшого моделювання нижчих рівнів декомпозиції.

У моделі IDEF0 відображено основні процеси, такі як створення та обробка заявок, а також управління користувачами та категоріями заявок. Для кожного процесу визначені вхідні та вихідні дані, а також механізми, що забезпечують їх виконання.

Нотація IDEF3 (Integration Definition for Process Description Capture) використовується для моделювання послідовності виконання процесів. Ця

нотація дозволяє деталізувати часовий порядок виконання функцій, що особливо важливо для опису бізнес-процесів технічної підтримки клієнтів.

У системі технічної підтримки клієнтів діаграма IDEF3 демонструє послідовність дій від моменту створення заявки до її обробки та закриття.

Основними елементами діаграми є:

- Процеси – описують конкретні дії, що виконуються.
- Зв'язки – визначають послідовність виконання процесів.
- Умови – умови, які впливають на виконання певних дій.

У моделі IDEF3 описано етапи, які проходить заявка, зокрема створення заявки клієнтом, аналіз заявки інженером технічної підтримки, фіксація рішення та зміна статусу заявки та закриття заявки.

Зв'язки між етапами демонструють логічний порядок виконання дій, а також можливі альтернативні сценарії, наприклад, повернення заявки клієнту для додаткової інформації.

Використання нотацій IDEF0 та IDEF3 у процесі моделювання інформаційної системи забезпечує наступні переваги. Забезпечується визначення чіткої структури функцій і послідовності дій. Ідентифікація вузьких місць у процесах. Полегшує розробку програмного забезпечення, забезпечуючи зрозумілу документацію для команди розробників.

Таким чином, створення діаграм у нотаціях IDEF0 та IDEF3 є невід'ємною частиною процесу розробки системи технічної підтримки клієнтів, сприяючи структурованому підходу до моделювання та оптимізації бізнес-процесів.

РОЗДІЛ 3.

РЕЗУЛЬТАТИ ПРОЄКТУВАННЯ ОСНОВНИХ МОДУЛІВ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТЕХПІДТРИМКИ КЛІЄНТІВ ТА РОЗРОБКА ІНТЕРФЕЙСУ КОРИСТУВАЧІВ WINDOWS FORMS

3.1. Доцільність проєктування основних модулів інформаційної системи технічної підтримки клієнтів та переваги інтерфейсу користувачів Windows Forms

Сучасні інформаційні системи технічної підтримки забезпечують швидку та ефективну відповідь запитів користувачів. Ефективна взаємодія між клієнтами, інженерами технічної підтримки та адміністраторами можлива лише за умови створення спроектованої архітектури системи, яка включає як модулі для обробки інформації, так і зручний інтерфейс користувача.

У цьому розділі представлені результати систем проєктування основних модулів технічної підтримки клієнтів, які забезпечують автоматизацію основних процесів – прийому заявок, їх обробки, управління користувачами та категоріями заявок. Крім того, у розділі буде відображено розробку інтерфейсу користувачів Windows Forms, яка є важливою складовою для забезпечення зручності та ефективності роботи користувачів із системою.

Проєктування модулів інформаційної системи є першим етапом у процесі її створення, оскільки саме модулі забезпечують основні функції системи. У системі технічної підтримки клієнтів такі модулі дозволяють автоматизувати обробку заявок, вести облік клієнтів і забезпечувати актуальність баз даних.

Ретельно спроектовані модулі дають можливість скоротити час обробки заявок завдяки чіткій структурі даних та автоматизованому виконанню операцій. Забезпечити придатність даних, що зберігаються в базі даних, та уникнути помилок через людський фактор. Полегшити систему обслуговування та її адаптацію до нових вимог за допомогою модульної архітектури. Крім того, модуль прийому заявок є основною системою, оскільки саме через нього клієнти

надають свої запити. Модуль обробки заявок дозволяє інженерам технічної підтримки швидко отримувати інформацію про проблему, переглядати прикріплені файли та змінювати статус заявки. Модулі управління користувачами та категоріями заявок забезпечують функціональність, необхідну для адміністрування систем.

Інтерфейс користувача є одним із ключових аспектів, які забезпечують успішність використання інформаційної системи. У випадку системи технічної підтримки зручний інтерфейс дозволяє користувачам ефективно взаємодіяти із системою, виконуючи завдання з мінімальними витратами часу.

Інтерфейс, розроблений на базі Windows Forms, має низку переваги (рис. 3.1).



Рисунок 3.1 – Переваги інтерфейсу користувача розробленого на базі Windows Forms

Зручність забезпечує інтуїтивно зрозуміле розташування елементів дозволяє користувачам швидко знаходити потрібні функції.

Функціональність дає можливість інтеграції різних елементів, таких як текстові поля, кнопки, списки та графічні компоненти, забезпечує повну реалізацію функціональних вимог.

Гнучкість інтерфейсу можна адаптувати під різні користувачі ролі (клієнти, інженери технічної підтримки, адміністратори).

Ефективність забезпечує оптимізовану взаємодію з базою даних через форму дозволяє забезпечити високу швидкість виконання операцій.

Розробка інтерфейсу Windows Forms для клієнтів системи технічної підтримки вимагає створення окремих форм для клієнтів, інженерів технічної підтримки та адміністраторів. Такий підхід дозволяє врахувати спеціальні потреби кожної ролі, забезпечуючи функціональність та зручність роботи.

Інтеграція розроблених модулів з інтерфейсом користувача дозволяє створити цілісну систему, яка відповідає потребам усіх сторінок сайту. Усі компоненти, включаючи обробку заявок, управління користувачами та категоріями заявок, об'єднані в єдиний інтерфейс, що забезпечує ефективну взаємодію між користувачами та системою.

3.2 Вибір засобів для проектування основних модулів інформаційної системи техпідтримки клієнтів та розробки інтерфейсу користувачів Windows Forms

Для успішного проектування інформаційної системи технічної підтримки клієнтам необхідно вибрати відповідні інструменти, які забезпечують ефективну реалізацію як основних модулів, так і зручного інтерфейсу користувача. Вибір засобів обґрунтовується на їх функціональності, інтеграційних можливостях, доступності та здатності забезпечити відповідність вимогам проекту (табл. 3.1).

Основною мовою програмування для реалізації проекту обрано C#. Ця мова є основним інструментом для створення додатків Windows Forms завдяки її тісній інтеграції з .NET Framework. Використання C# забезпечує зручний синтаксис, широкий вибір бібліотек для роботи з базами даних, мережевими запитами та графічними елементами. Крім того, C# дозволяє створювати

інтерактивний та інтуїтивно зрозумілий інтерфейс користувача з мінімальними витратами часу.

Таблиця 3.1 – Вибрані засоби для створення основних модулів інформаційної системи техпідтримки клієнтів

Засіб	Значення в інформаційній системі
C#	Основна мова програмування для розробки модулів та графічного інтерфейсу користувача (Windows Forms).
SQL Server	Система управління базами даних для зберігання запитів, користувачів, категорій та інших даних.
Windows Forms	Інструмент для створення графічного інтерфейсу, включаючи форми введення даних, кнопки, списки та інші елементи управління.
Visual Studio	Інтегроване середовище розробки для створення, тестування, налагодження та управління проектом.
dbdiagram.io	Інструмент для моделювання баз даних, створення концептуальних і логічних моделей, які відображають зв'язки між таблицями.
ADO.NET/Entity Framework	Бібліотеки для інтеграції C# із базою даних SQL Server, виконання запитів і обробки даних.
UML-діаграми	Стандарт для створення діаграми, що описує функціональні можливості та взаємодію модулів системи.

Для збереження даних у системі вибрано SQL Server як систему керування базами даних (СКБД). SQL Server забезпечує високу продуктивність у роботі з великими обсягами даних, підтримує розширені функції безпеки та легко інтегрується з додатками .NET через бібліотеки ADO.NET або Entity Framework. Вибір цього СКБД зумовлений її стабільністю, масштабістю та широкими можливостями для обробки структурованих даних, що є місцем для управління заявками клієнтів.

Для проектування бази даних використовують засоби візуального моделювання, такі як `dbdiagram.io`, які можуть створити концептуальні та логічні моделі баз даних. Цей інструмент забезпечує простоту у візуалізації зв'язків між таблицями та генерує код для подальшого використання в СКБД. Також для створення діаграми використовується стандарт UML, який відображає функціональні можливості модулів.

Графічний інтерфейс користувача розробляється з використанням `Windows Forms`, що є частиною `.NET Framework`. Інструмент забезпечує простоту у створенні форм, кнопок, текстових полів та інших елементів, які використовують для взаємодії користувача з системою. `Windows Forms` дозволяє швидко створювати додатки, які інтегруються з базою даних та іншими компонентами системи.

Для тестування та налагодження системи обрано інтегроване середовище розробки `Visual Studio`. Це середовище є стандартом для створення `.NET`-додатків і забезпечує великий набір інструментів для відлагодження, профілювання коду та створення модульних тестів. `Visual Studio` також надає можливість зручної роботи з `Windows Forms`, а її вбудовані інструменти не дозволяють мінімізувати час розробки.

Таким чином, для реалізації проекту використовуються поєднання `C#`, `SQL Server`, `Windows Forms` та `Visual Studio`, що забезпечує ефективність розробки, інтеграцію між модулями та стабільність системи. Вибрані засоби можуть створити продуктивну інформаційну систему, яка відповідає сучасним вимогам до програмного забезпечення та забезпечує високу якість обслуговування клієнтів.

3.3. Проектування основних модулів інформаційної системи

3.3.1. Модуль прийому заявок

Модуль прийому заявок є центральною частиною інформаційної системи технічної підтримки клієнтів. Він забезпечує зручний спосіб для клієнтів подати заявку, надаючи всю необхідну інформацію для подальшої обробки інженерами. Основне завдання цього модуля – автоматизація процесу створення заявок і забезпечення коректного збереження даних у базі.

Модуль прийому заявок дозволяє клієнтам:

- заповнювати заявку з використанням інтерактивної форми, яка включає заголовок, опис проблеми та можливість додавання файлів;
- перевіряти коректність введених даних через вбудовані механізми валідації;
- отримувати унікальний ідентифікатор заявки та повідомлення про успішну її реєстрацію;
- інженери, у свою чергу, отримують доступ до бази даних, де автоматично зберігаються всі створені заявки.

Процес починається із заповнення форми клієнтом. Після підтвердження введених даних інформація з форми передається на сервер, де проходить додаткову валідацію. Далі дані записуються в базу даних. Після успішного додавання заявки клієнт отримує повідомлення про реєстрацію. Фрагмент коду на мові C# модуля прийому заявок подано на рис. 3.2.


```

{
    // Перевірка заповнення обов'язкових полів
    if (string.IsNullOrEmpty(headline) || string.IsNullOrEmpty(description))
    {
        MessageBox.Show("Будь ласка, заповніть усі обов'язкові поля.");
        return;
    }

    // Підготовка SQL-запиту
    string query = @"INSERT INTO ticket (headline, description, creation_date, user_id, type_id, status)
    VALUES (@headline, @description, GETDATE(), @userId, @typeId, 'Відкрита')";

    using (SqlConnection connection = new SqlConnection(connectionString))
    using (SqlCommand command = new SqlCommand(query, connection))
    {
        command.Parameters.AddWithValue("@headline", headline);
        command.Parameters.AddWithValue("@description", description);
        command.Parameters.AddWithValue("@userId", userId);
        command.Parameters.AddWithValue("@typeId", typeId);

        // Виконання запиту
        connection.Open();
        command.ExecuteNonQuery();
    }

    MessageBox.Show("Заявку успішно створено!");
}

```

Рисунок 3.2 – Фрагмент коду модуля прийому заявок

Введені дані перевіряються на заповнення. Якщо обов'язкові поля не заповнені, клієнт отримує повідомлення про помилку.

Дані передаються у SQL-запит із використанням параметрів для захисту від SQL-ін'єкцій. Функція `ExecuteNonQuery()` записує інформацію до бази даних, після чого клієнту виводиться повідомлення про успіх.

Логічна модель таблиці `ticket`, яка використовується для збереження заявок, наведена у таблиці 3.2.

Таблиця 3.2 – Структура таблиці `ticket`

Назва поля	Тип даних	Опис
<code>ticket_id</code>	INT (PK)	Унікальний ідентифікатор заявки
<code>headline</code>	VARCHAR(255)	Заголовок заявки
<code>description</code>	TEXT	Детальний опис проблеми
<code>creation_date</code>	DATETIME	Дата та час створення заявки
<code>user_id</code>	INT (FK)	Зовнішній ключ до таблиці <code>user</code>
<code>type_id</code>	INT (FK)	Зовнішній ключ до таблиці <code>type</code>
<code>status</code>	VARCHAR(50)	Статус заявки (Відкрита/Закрита)

Передбачається використовувати SQL-запити для управління даними. Зокрема, для перегляду усіх заявок у системі використовується наступний SQL-запит, фрагмент коду якого представлено на рис. 3.3.

```
SELECT ticket_id, headline, description, creation_date, status
FROM ticket
WHERE user_id = @userId
ORDER BY creation_date DESC;
```

Рисунок 3.3 – Фрагмент коду SQL-запиту для перегляду усіх заявок у системі

Вибираються лише ті заявки, які створені певним користувачем (@userId), та сортуються за датою створення у порядку спадання.

Модуль прийому заявок є базовим компонентом інформаційної системи технічної підтримки. Він забезпечує ефективний прийом та збереження даних, зменшуючи ризик втрати інформації та оптимізуючи подальшу обробку заявок.

3.3.2. Модуль обробки заявок

Модуль обробки заявок є центральною частиною інформаційної системи технічної підтримки, що забезпечує функціональність для інженерів. Основне завдання цього модуля – надання доступу до заявок, можливість перегляду їх вмісту, фіксація рішень та зміна статусу.

Інженери технічної підтримки мають доступ до списку заявок із можливістю їх перегляду, сортування та фільтрації за статусом або іншими параметрами. Кожна заявка містить інформацію про її заголовок, опис, дату створення, статус і прикріплені файли. Інженери також можуть залишати коментарі та фіксувати рішення, ускладнені для вирішення проблеми.

Інтерфейс включає список заявок із функціями пошуку та фільтрації, форму для перегляду деталей заявки, а також поле для внесення коментарів і вибору нового статусу.

Процес зміни статусу заявки та фіксації результату виглядає наступним чином. Після аналізу заявки інженер може оновити її статус, наприклад, із «Відкрита» на «Закрита», якщо проблема вирішена. При зміні статусу також додається коментар або виконаних дій. Інформація передається в базу даних, де зберігаються оновлення, що забезпечує прозорість і можливість відстеження історії заявки.

Фрагмент коду зміни статусу заявки представлено на рис. 3.4.

```
{
// Перевірка обов'язкових полів
if (string.IsNullOrEmpty(newStatus) || string.IsNullOrEmpty(engineerComment))
{
    MessageBox.Show("Будь ласка, заповніть статус і коментар.");
    return;
}

// Підготовка SQL-запиту
string query = @"UPDATE ticket
                SET status = @newStatus,
                    engineer_comment = @engineerComment,
                    last_updated = GETDATE()
                WHERE ticket_id = @ticketId";

using (SqlConnection connection = new SqlConnection(connectionString))
using (SqlCommand command = new SqlCommand(query, connection))
{
    command.Parameters.AddWithValue("@newStatus", newStatus);
    command.Parameters.AddWithValue("@engineerComment", engineerComment);
    command.Parameters.AddWithValue("@ticketId", ticketId);

    // Виконання запиту
    connection.Open();
    command.ExecuteNonQuery();
}
}
```

Рисунок 3.4 – Фрагмент коду зміни статусу заявки

Перевіряються обов'язкові поля для статусу і коментаря. SQL-запит оновлює статус заявки, додає коментар і оновлює дату останньої зміни. Успішне виконання запиту завершується виведенням повідомлення для інженера.

Модуль обробки заявок інтегрується з базою даних, використовуючи таблицю ticket, яка зберігає всю інформацію про заявки (табл. 3.3).

Таблиця 3.3 – Структура таблиці ticket для обробки заявок

Назва поля	Тип даних	Опис
ticket_id	INT (PK)	Унікальний ідентифікатор заявки
headline	VARCHAR(255)	Заголовок заявки
description	ТЕКСТ	Опис проблеми
status	VARCHAR(50)	Статус заявки (Відкрита/Закрита)
engineer_comment	ТЕКСТ	Коментар інженера
last_updated	DATETIME	Час останнього оновлення

Для перегляду списку заявок інженером слід використовувати наступний SQL-запит, який представлено на рис. 3.5.

```
SELECT ticket_id, headline, status, last_updated  
FROM ticket  
WHERE status = 'Відкрита'  
ORDER BY last_updated DESC;
```

Рисунок 3.5 – Фрагмент коду SQL-запиту для перегляду списку заявок інженером

Вибираються лише заявки зі статусом «Відкрита». Результати сортуються за датою останнього оновлення, щоб інженер працював із найактуальнішими заявками.

Алгоритм обробки заявки включає такі етапи:

- 1) Перегляд заявки в деталях;
- 2) Аналіз проблеми та визначення рішення;
- 3) Зміна статусу заявки, внесення коментаря;
- 4) Оновлення інформації в базі даних.

Модуль обробки заявок є невід’ємною частиною системи, що дозволяє інженерам ефективно працювати із заявками клієнтів. Завдяки інтеграції з базою даних і зручному інтерфейсу цей модуль забезпечує прозорість процесів і підвищену ефективність технічної підтримки.

3.3.3. Модуль управління користувачами

Модуль управління користувачами є вагомю частиною інформаційної системи технічної підтримки клієнтів. Він забезпечує інструментами адміністратора для створення нових облікових записів, редагування даних користувачів та їх видалення. Цей модуль також відповідає безпеці доступу до системи через механізми автентифікації та управління ролями.

Існують наступні можливості адміністратора щодо додавання, редагування та видалення користувачів. Адміністратор має доступ до списку всіх систем користувачів за допомогою:

- додавання нового користувача – включає введення основних даних (логін, пароль, контактна інформація, роль). нові облікові записи автоматично додаються в базу даних із зазначенням унікального ідентифікатора;
- редагування існуючих користувачів – можливість змінити дані користувача, таку як контактну інформацію чи роль (наприклад, змінити роль клієнта на адміністратора);
- видалення користувачів – функція видалення користувачів, які більше не мають доступу до системи.

Фрагмент коду для додавання нового користувача подано на рис. 3.6.

```

{
    // Перевірка заповнення обов'язкових полів
    if (string.IsNullOrEmpty(username) || string.IsNullOrEmpty(password) || string.IsNullOrEmpty(role))
    {
        MessageBox.Show("Будь ласка, заповніть  всі обов'язкові поля.");
        return;
    }

    // Підготовка SQL-запиту
    string query = @"INSERT INTO users (username, password, email, role)
    | | | | VALUES (@username, HASHBYTES('SHA2_256', @password), @email, @role)";

    using (SqlConnection connection = new SqlConnection(connectionString))
    using (SqlCommand command = new SqlCommand(query, connection))
    {
        command.Parameters.AddWithValue("@username", username);
        command.Parameters.AddWithValue("@password", password);
        command.Parameters.AddWithValue("@email", email);
        command.Parameters.AddWithValue("@role", role);

        // Виконання запиту
        connection.Open();
        command.ExecuteNonQuery();
    }

    MessageBox.Show("Користувача успішно додано!");
}

```

Рисунок 3.6 – Фрагмент коду для додавання нового користувача

Поля перевіряються на заповнення. Пароль зберігається в хешованому вигляді для підвищення безпеки. SQL-запит додає дані нового користувача до бази.

Модуль управління користувачами включає функції забезпечення безпеки. Зокрема передбачається хешування паролів. Усі паролі зберігаються в хешованому вигляді з використанням алгоритмів SHA-256. Це включає можливість зчитування паролів у разі доступу до бази даних.

Окрім того, передбачається розподіл ролей. Кожен користувач має роль (клієнт, інженер, адміністратор), яка забезпечує рівень доступу до функціональної системи. Механізм аутентифікації передбачає, що користувач повинен ввести правильний логін і пароль для доступу до системи. Система реєструє дії адміністратора, такі як додавання чи видалення користувачів, для забезпечення прозорості.

Фрагмент коду для перевірки аутентифікації подано на рис. 3.7.

```

{
    string query = @"SELECT COUNT(*) FROM users
                    WHERE username = @username AND password = HASHBYTES('SHA2_256', @password)";
    using (SqlConnection connection = new SqlConnection(connectionString))
    using (SqlCommand command = new SqlCommand(query, connection))
    {
        command.Parameters.AddWithValue("@username", username);
        command.Parameters.AddWithValue("@password", password);

        connection.Open();
        int result = (int)command.ExecuteScalar();

        return result > 0;
    }
}

```

Рисунок 3.7 – Фрагмент коду для перевірки аутентифікації

SQL-запитує перевірку наявності користувача із заданим логіном і хешованим паролем. У разі успіху повертається true, що дозволяє користувачу увійти до системи.

Логічна структура таблиці users представлена у таблиці 3.4.

Таблиця 3.4 – Структура таблиці users для управління користувачами

Назва поля	Тип даних	Опис
user_id	INT (PK)	Унікальний ідентифікатор користувача
username	VARCHAR(100)	Вхід користувача
password	VARBINARY(MAX)	Хешований пароль
email	VARCHAR(255)	Контактна електронна пошта
role	VARCHAR(50)	Роль користувача (Клієнт/Адміністратор/Інженер)

Модуль управління користувачами дозволяє адміністраторам ефективно керувати системами користувачів, підтримуючи її безпеку та прозорість. Завдяки функції автентифікації та журналу дій забезпечується високий рівень захисту інформації та контроль доступу до системи.

3.3.4. Модуль управління категоріями заявок

Модуль управління категоріями заявок є кількістю компонентів інформаційної системи технічної підтримки клієнтів, що дозволяє адміністраторам створювати, редагувати та видаляти типи заявок. Це забезпечує класифікацію заявок, що спрощує їх обробку інженерами технічної підтримки.

Адміністратор може створювати нові типи заявок, які починають їх перевагу та специфіку. Наприклад, заявки можуть бути класифіковані як «Термінова» або «Стандартна». Редагування типів заявок дозволяє змінити їх опис вимог або пріоритетність у разі зміни системи.

Фрагмент коду для створення нового типу заявки подано на рис. 3.8.

```

{
    // Перевірка обов'язкових полів
    if (string.IsNullOrEmpty(typeName) || string.IsNullOrEmpty(typeDescription))
    {
        MessageBox.Show("Будь ласка, заповніть  всі обов'язкові поля.");
        return;
    }

    // Підготовка SQL-запиту
    string query = @"INSERT INTO type (type_name, type_description)
                    VALUES (@typeName, @typeDescription)";

    using (SqlConnection connection = new SqlConnection(connectionString))
    using (SqlCommand command = new SqlCommand(query, connection))
    {
        command.Parameters.AddWithValue("@typeName", typeName);
        command.Parameters.AddWithValue("@typeDescription", typeDescription);

        // Виконання запиту
        connection.Open();
        command.ExecuteNonQuery();
    }

    MessageBox.Show("Тип заявки успішно додано!");
}

```

Рисунок 3.8 – Фрагмент коду для створення нового типу заявки

Перевіряється заповнення полів, щоб уникнути збереження неповної інформації. SQL-запит додає новий тип заявки до бази даних із зазначенням її назви та опису.

Передбачається актуалізація категорій у базі даних. Модуль управління категоріями забезпечення актуальності даних у базі. Адміністратор може змінити опис існуючих категорій або видалити ті, які більше не використані. Усі зміни відображаються в базі даних у реальному часі, що забезпечує придатність класифікації заявок.

Фрагмент коду для оновлення типу заявки подано на рис. 3.9.

```

{
    // Перевірка обов'язкових полів
    if (string.IsNullOrEmpty(newTypeName) || string.IsNullOrEmpty(newTypeDescription))
    {
        MessageBox.Show("Будь ласка, заповніть усі обов'язкові поля.");
        return;
    }

    // Підготовка SQL-запиту
    string query = @"UPDATE type
                    SET type_name = @newTypeName,
                        type_description = @newTypeDescription
                    WHERE type_id = @typeId";

    using (SqlConnection connection = new SqlConnection(connectionString))
    using (SqlCommand command = new SqlCommand(query, connection))
    {
        command.Parameters.AddWithValue("@newTypeName", newTypeName);
        command.Parameters.AddWithValue("@newTypeDescription", newTypeDescription);
        command.Parameters.AddWithValue("@typeId", typeId);

        // Виконання запиту
        connection.Open();
        command.ExecuteNonQuery();
    }

    MessageBox.Show("Тип заявки успішно оновлено!");
}

```

Рисунок 3.9 – Фрагмент коду для оновлення типу заявки

Запит оновлює назву та опис типу заявки на основі її ідентифікатора. Дані передаються як параметри, що завершили безпеку запиту.

Логічна структура таблиці type представлена у табл. 3.5.

Таблиця 3.5 – Структура таблиці type для категорій заявок

Назва поля	Тип даних	Опис
type_id	INT (PK)	Унікальний ідентифікатор типу
type_name	VARCHAR(100)	Назва типу заявки
type_description	ТЕКСТ	Опис типу заявки

Модуль управління категоріями заявок є ключовим інструментом для адміністратора, який забезпечує актуальність і зручність класифікації заявок. Завдяки цьому модулю система адаптується до потреб клієнтів, а інженери підтримують чітко структуровані заявки для ефективної обробки.

3.4. Розробка інтерфейсу користувачів Windows Forms

Розробку форм для інформаційної системи технічної підтримки клієнтів виконано за допомогою Windows Forms у середовищі розробки Microsoft Visual Studio. Для цього створюється проект «Windows Forms Application (.NET Framework)», виконується необхідне налаштування, після чого отримуємо середовище для проектування інтерфейсу користувача програми (рис. 3.10).

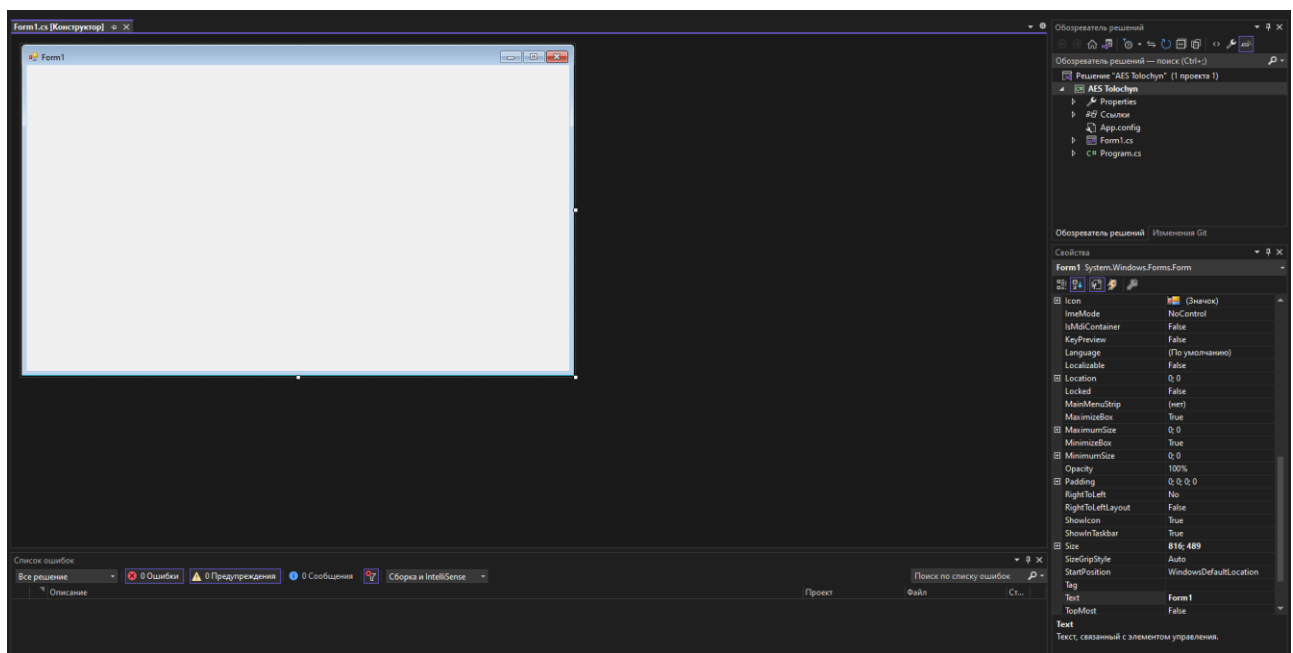


Рисунок 3.10 – Налаштування середовища розробки Microsoft Visual Studio

Після створення проекту, за допомогою стандартних компонентів Windows Forms розробляються необхідні поля для введення та редагування даних, а також кнопки для керування програмою. У проекті використовувалися елементи, такі як RichTextBox (для введення та редагування текстових даних), Button (для взаємодії з функціоналом програми), TextBox (для введення

текстової інформації) та Label (для підписів до полів). Зовнішній вигляд програмного забезпечення з базовим налаштуванням цих елементів представлено на рис. 3.11.

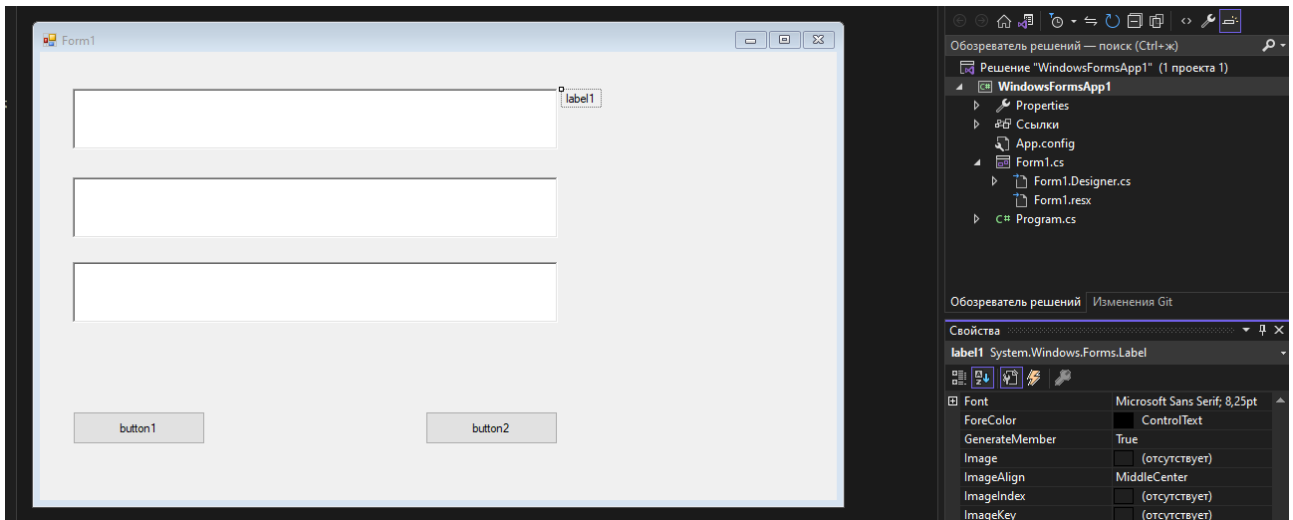


Рисунок 3.11 – Попередній вигляд інтерфейсу клієнта для створення заявки без налаштування елементів

Розробка інтерфейсу користувачів є ключовим етапом у створенні інформаційної системи технічної підтримки клієнтів. Інтерфейс на основі Windows Forms забезпечує простоту у використанні та високу інтерактивність, що дозволяє ефективно взаємодіяти з системою. В рамках цього розділу було створено три основні форми – для клієнтів, інженерів технічної підтримки та адміністраторів.

Інтерфейс клієнта дозволяє подавати заявки на вирішення технічних проблем. Форма включає такі елементи:

- поля для введення заголовка та опису заявки;
- кнопку для додавання прикріплених файлів (скріншоти, логи);
- кнопку для надсилання заявки до системи.

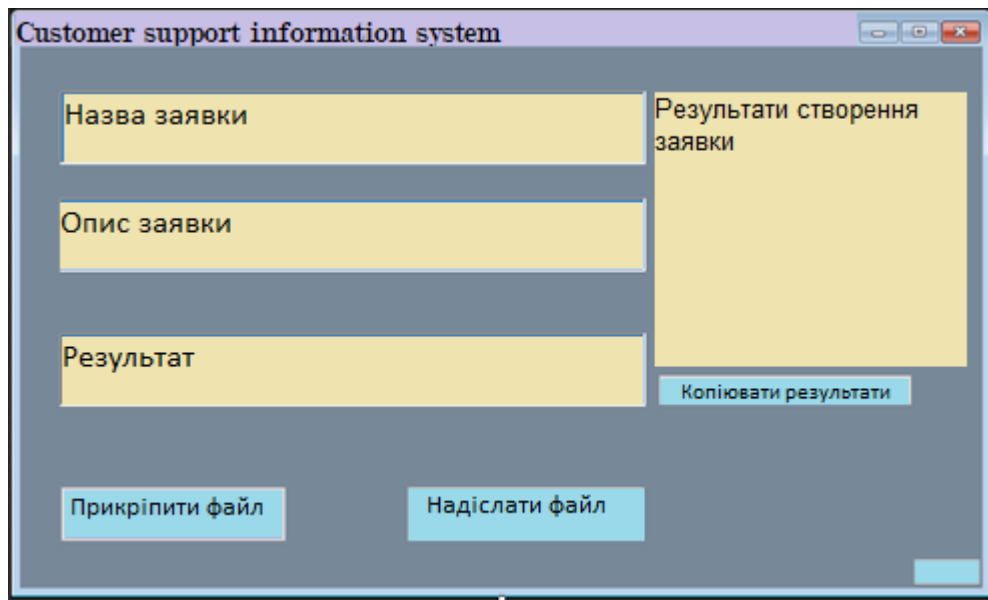


Рисунок 3.12 – Інтерфейс клієнта для створення заявки

Форма також виводить повідомлення про успішне створення заявки, що клієнту впевниться, що інформація була отримана.

Інтерфейс інженера технічної підтримки має свою форму. Форма для інженерів містить:

- список усіх заявок із можливістю фільтрації за статусом («відкрита», «закрита»);
- поле для перегляду деталей вибраної заявки, включаючи заголовок, опис і прикріплені файли;
- поле для введення коментаря та вибору нового статусу;
- кнопку для збереження змін у базі даних.

Цей інтерфейс дозволяє швидко аналізувати заявки та фіксувати результати їх обробки.

Інтерфейс адміністратора також має свою форму. Інтерфейс адміністратора дозволяє керувати користувачами та категоріями заявок. Основні функції:

- додавання, редагування та видалення облікових записів користувачів;
- створення нових категорій заявок або редагування існуючих;

➤ відображення користувачів і категорій із можливістю пошуку та фільтрації.

Форма для адміністратора побудована таким чином, щоб забезпечити максимальну зручність роботи з великими обсягами даних.

У додатку А наведено приклад коду, який реалізує основну форму Windows Forms із меню навігації між ролями користувачів.

Усі елементи інтерфейсу інтуїтивно зрозумілі, що забезпечують легке використання системи. Інтерфейси адаптовані під потреби кожного користувача. Оптимізований код дозволяє швидко виконувати всі операції.

Розроблений інтерфейс Windows Forms забезпечує ефективну взаємодію між клієнтами, інженерами та адміністраторами, що сприяє високій продуктивності системи.

РОЗДІЛ 4.

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Аналіз небезпек та шкідливих виробничих під час розробки інформаційної системи

Розробка інформаційної системи (ІС) є складним процесом, який включає в себе виконання різноманітних робіт, таких як проектування, програмування, тестування та впровадження. Під час виконання цих робіт можуть виникати небезпеки та шкідливі виробничі фактори, які можуть призвести до травм, захворювань або інших негативних наслідків для здоров'я та безпеки працівників.

Перелік можливих небезпек, які можуть виникати під час розробки ІС, представлено в таблиці 4.1.

Таблиця 4.1 – Перелік можливих небезпек, які можуть виникати під час розробки ІС

Вид небезпеки	Характеристика небезпеки
Механічні небезпеки	травми, пов'язані з обладнанням, інструментами та матеріалами, які використовуються під час розробки ІС
Електричні небезпеки	травми, пов'язані з використанням електричного обладнання та інструментів
Хімічні небезпеки	травми, пов'язані з використанням хімічних речовин, які можуть бути присутніми в ІС або в матеріалах, які використовуються для її розробки.
Психофізіологічні небезпеки	стрес, перенапруження, втома, які можуть виникати внаслідок тривалої роботи за комп'ютером або в умовах високих вимог до якості та дедлайнів

При цьому можуть виникати шкідливі виробничі фактори. До шкідливих виробничих факторів, які можуть впливати на здоров'я працівників під час розробки ІС, належать:

- забруднення повітря – пил, гази, пари, які можуть виділятися при роботі з обладнанням, інструментами та матеріалами;
- забруднення шумом – шум, який може створюватися обладнанням, інструментами та іншими джерелами;
- забруднення освітленням – недостатнє або надмірне освітлення робочого місця;
- забруднення мікрокліматом – несприятливі умови мікроклімату робочого місця, такі як підвищена або знижена температура, вологість, загазованість.

4.2. Розробка заходів щодо покращення умов праці

Нами запропоновано заходи щодо запобігання можливих небезпек, які можуть виникати під час розробки ІС. Для запобігання небезпекам та шкідливим виробничим факторам під час розробки ІС необхідно вжити таких заходів. Забезпечити належний стан обладнання, інструментів та матеріалів, які використовуються під час розробки ІС.

Організувати навчання працівників з питань охорони праці та безпеки. Забезпечити дотримання норм і правил охорони праці та безпеки на робочому місці.

Заходи щодо запобігання небезпекам та шкідливим виробничим факторам під час розробки ІС можуть включати в себе:

- використання безпечного обладнання та інструментів. Наприклад, для роботи з електричним обладнанням необхідно використовувати захисні засоби, такі як діелектричні рукавички та боти;

- використання безпечних матеріалів. Наприклад, при роботі з хімічними речовинами необхідно використовувати захисний одяг та респіратор;
- організація робочого місця відповідно до вимог охорони праці та безпеки. Наприклад, робоче місце необхідно забезпечити достатнім освітленням та вентиляцією;
- проведення регулярних медичних оглядів працівників.

Впровадження таких заходів дозволить підвищити рівень охорони праці та безпеки під час розробки ІС, а також знизити ризик травм, захворювань та інших негативних наслідків для здоров'я та безпеки працівників.

4.3. Розробка логічно-імітаційної моделі процесу виникнення травм під час монтажу інтелектуальної інформаційної системи

Для побудови логіко-імітаційної моделі процесу, формування і виникнення аварії та травми під час монтажу інформаційної системи складемо список базових подій. Вони лежатимуть у основі даної моделі. Кожному пункту списку присвоюємо певне значення ймовірності виникнення. Нижче подано сам список:

- | | |
|--|-------------------------------------|
| 1. Стан контролю з охорони праці | $P_1 = 0,2;$ |
| 2. Несерйозне відношення до проходження ТО інструменту | $P_2 = 0,1;$ |
| 3. Відсутність комплектуючих установки..... | $P_3 = 0,2;$ |
| 4. Невисока міцність | $P_4 = 0,03;$ |
| 1. Використання застарілого обладнання..... | $P_6 = 0,02;$ |
| 2. Попадання сторонніх предметів | $P_7 = 0,4;$ |
| 3. Досвід роботи виконавця | $P_{12} = 0,35.$ |
| 4. Професійний рівень виконавця | $P_{13} = 0,5;$ |
| 5. Психофізіологічний стан виконавця..... | $P_{14} = 0,083;$ |

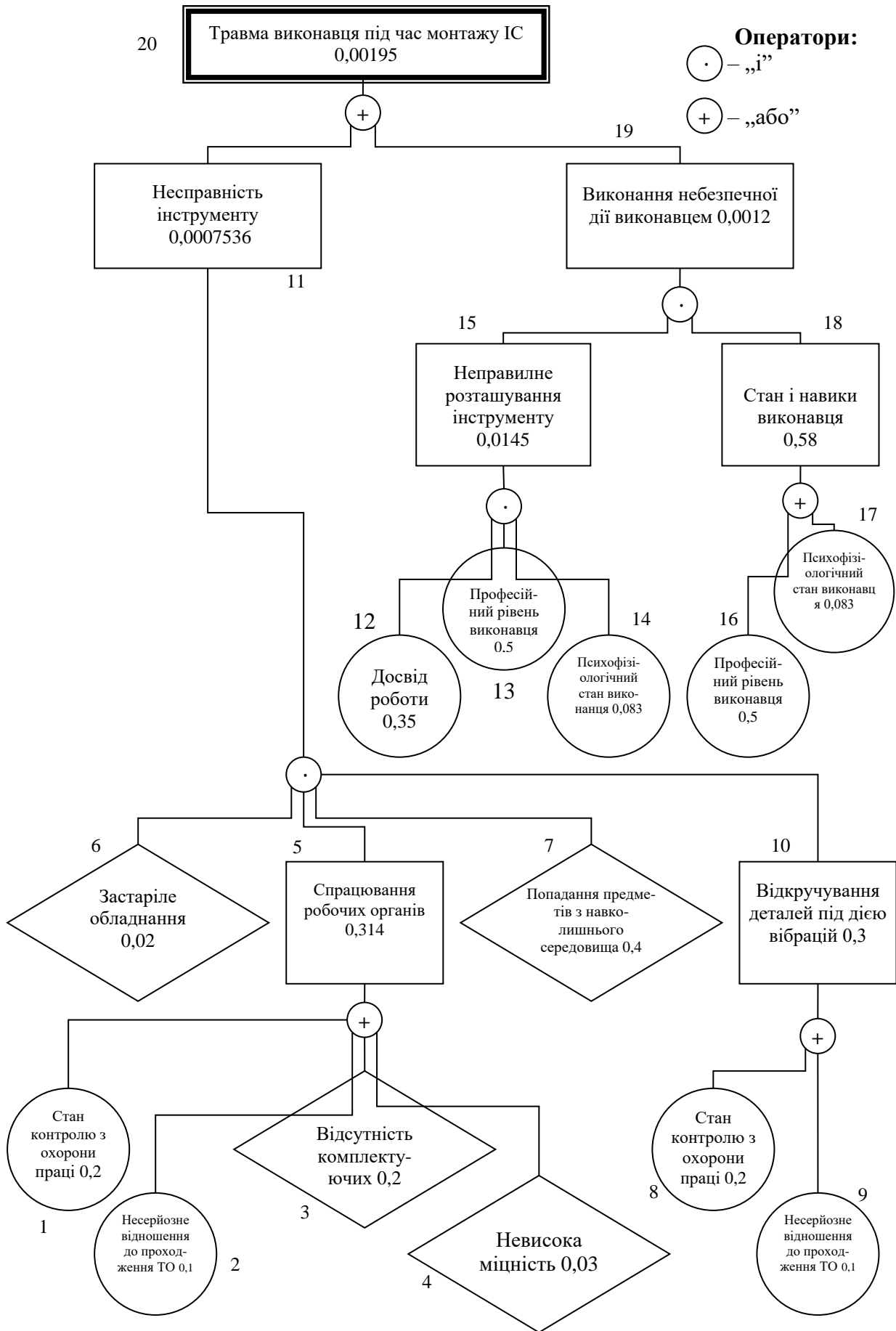


Рис. 4.1. Логіко-імітаційна модель процесу формування та виникнення аварії та травми під час монтажу інформаційної системи

На основі даного списку будуюмо матрицю логічних взаємозв'язків між окремими пунктами, графічне представлення якої зображено на рис. 4.1.

Розрахуємо ймовірності виникнення подій, що входять у дану логіко-імітаційну модель процесу монтажу інформаційної системи (на прикладі ймовірності отримання травми виконавця).

Ймовірність виникнення події P_5 визначаємо наступним чином:

$$P_5 = 0,2 + 0,1 + 0,2 + 0,003 - 0,2 \cdot 0,1 - 0,2 \cdot 0,03 - 0,2 \cdot 0,03 - 0,1 \cdot 0,2 - 0,1 \cdot 0,03 - 0,2 \cdot 0,03 + 0,2 \cdot 0,1 \cdot 0,2 + 0,1 \cdot 0,2 \cdot 0,03 + 0,2 \cdot 0,1 \cdot 0,2 + 0,2 \cdot 0,1 \cdot 0,03 - 0,2 \cdot 0,1 \cdot 0,2 \cdot 0,03 = 0,314$$

Ймовірність виникнення події P_{10} визначаємо так:

$$P_{10} = 0,2 + 0,1 = 0,3.$$

Ймовірність виникнення події P_{11} визначаємо:

$$P_{11} = 0,02 \cdot 0,314 \cdot 0,4 \cdot 0,3 = 0,00075.$$

Ймовірність виникнення події P_{15} визначаємо наступним чином:

$$P_{15} = 0,35 \cdot 0,5 \cdot 0,083 = 0,0145.$$

Ймовірність події P_{18} :

$$P_{18} = 0,5 + 0,083 = 0,58.$$

Ймовірність події P_{19} :

$$P_{19} = 0,0145 \cdot 0,083 = 0,0012.$$

Ймовірність події P_{20} :

$$P_{20} = 0,00075 + 0,0012 = 0,00195.$$

Ймовірність травми рівна ймовірності виникнення аварії, бо остання можлива лише за умови монтажу інформаційної системи людиною.

Після аналізу результатів моделювання ймовірність виникнення травми можна звести до дуже малої величини – достатньо зменшити вплив ймовірностей вихідних факторів, які до неї призводять.

4.4. Розробка заходів щодо безпеки у надзвичайних ситуаціях

Науково-технічний прогрес радикально змінив світ, породивши нові загрози для цивілізації. У житті сучасної людини все більше місце займають турботи, пов'язані з подоланням різних кризових явищ, що виникають в процесі розвитку земної цивілізації. В Україні, як і в усьому світі, в останні роки спостерігається зростання числа військових дій, катастроф природного та техногенного характеру. Це обумовлено, перш за все, прогресуючої урбанізацією територій, збільшенням щільності населення Землі, і, як наслідок, збільшенням антропогенного навантаження на навколишнє середовище. Захист природних систем і населення від надзвичайних ситуацій різного характеру сформувався в останні роки як нагальна і об'єктивна потреба суспільства і держави.

Заходи щодо захисту цивільного населення плануються проводяться по населених пунктах де розміщені підприємства і охоплюють населення навколишніх сіл. Водночас характер та зміст захисних засобів встановлюються від ступеня загрози, місцевих умов з урахуванням важливості виробництва для безпеки населення і інших економічних і соціальних чинників.

Основні заходи щодо захисту населення плануються та здійснюються завчасно і мають випереджувальний характер, це стосується насамперед підготовки, підтримання у постійній готовності індивідуальних та колективних засобів захисту, їх накопичення, а також підготовки до проведення евакуації населення із зон підвищеного ризику.

Також раз в три роки проводяться навчання по підготовці близьких до військових дій, що в разі небезпеки могло би не дістати людину зненацька. Керівництво докладає максимум зусиль, щоб працівники підприємств були хоча би мінімально захищенні в разі будь-якої небезпеки пов'язаної з тими чи іншими обставинами.

РОЗДІЛ 5.

ВИЗНАЧЕННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД ВИКОРИСТАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТЕХПІДТРИМКИ КЛІЄНТІВ

Розробка та впровадження інформаційної системи технічної підтримки клієнтів дозволяє значно скоротити витрати на обробку заявок, підвищити продуктивність праці працівників технічної підтримки та скоротити час обробки заявок. Нами проведено оцінку системи економічної ефективності за допомогою відповідних показників, формул, прикладів розрахунків та аналізу отриманих результатів.

Для визначення ефективності використовують систему показників. Економія витрат на обробку заявок (E_e) визначається за формулою:

$$E_e = C_o - C_n, \quad (5.1)$$

де C_o – витрати на обробку заявок до впровадження системи; C_n – витрати після впровадження системи.

Підвищення продуктивності (Π_n) визначається за формулою:

$$\Pi_n = \frac{H_o}{H_n}, \quad (5.2)$$

де H_o – кількість працівників до впровадження; H_n – кількість працівників після впровадження.

Скорочення часу (T_e) обробки замовлень визначається за формулою:

$$T_e = T_o - T_n, \quad (5.3)$$

де T_o – середній час обробки заявки до впровадження; T_n – середній час після впровадження.

Виконаємо визначення економічної ефективності від використання інформаційної системи техпідтримки клієнтів для наступних умов:

- витрати на обробку заявок до впровадження $C_o = 50000$ грн/місяць;

➤ витрати на обробку заявок після впровадження $C_n = 30000$ грн/місяць;

- кількість працівників до впровадження $H_0 = 10$ осіб;
- кількість працівників після впровадження $H_n = 6$ осіб;
- середній час обробки заявки до впровадження $T_0 = 1$ година;
- середній час після впровадження $T_n = 30$ хвилин.

Економія витрат на обробку заявок (E_g) становить:

$$E_g = 50000 - 30000 = 20000 \text{ грн/місяць.}$$

Підвищення продуктивності (P_n) становить:

$$P_n = \frac{10}{6} = 1.67 \text{ разів.}$$

Скорочення часу (T_e) обробки замовлень становить:

$$T_e = 60 - 30 = 30 \text{ хвилин / заявку.}$$

Результати виконаних розрахунків наведено у таблиці 5.1.

Таблиця 5.1 – Показники економічної ефективності інформаційної системи

Показник	До впровадження	Після впровадження	Результат
Витрати на обробку заявок (грн)	50 000	30 000	Економія на 20000 грн
Кількість працівників	10	6	Зменшення на 4 працівника
Час обробки заявки (хвилин)	60	30	Зростання швидкості на 30 хвилин
Продуктивність	–	–	Зростання у 1,67 разів

Результати показують значне зниження витрат на обробку заявок, що забезпечує економію 20000 грн на місяць. Зменшення кількості працівників від 10 до 6 дозволяє скоротити витрати на оплату праці, а скорочення часу обробки заявки на 30 хвилин продуктивності системи. Впровадження інформаційної системи дозволяє не тільки зменшити витрати, але й оптимізувати процеси технічної підтримки, забезпечуючи кращий сервіс для клієнтів.

ВИСНОВКИ І ПРОПОЗИЦІЇ

Актуальність теми роботи зумовлена зростаючою потребою в існуючому процесі технічної підтримки. Ручне управління заявками та комунікаціями із клієнтами часто призводить до помилок, затримок і незадоволеності користувачів. Впровадження автоматизованої інформаційної системи вирішує ці проблеми, забезпечуючи централізоване управління запитами, швидкий доступ до історії обслуговування клієнтів, а також підвищуючи прозорість і підконтрольність процесів.

Нами виконано аналіз існуючого стану техпідтримки клієнтів організацій. На основі аналізу компаній, які активно працюють у сфері технічної підтримки, можна виділити три основні підходи, які вони використовують (рис. 1.1) – традиційний підхід, автоматизовані системи технічної підтримки клієнтів організацій та гібридний підхід.

Нами виконано аналіз існуючих інструментів технічної підтримки, які представлено у табл. 1.1. На рисунку 1.2 показано типовий інтерфейс системи Zendesk, який демонструє функціональне управління запитами, включаючи статус перегляду, пріоритету та коментарів до заявок.

Zoho Mail Desktop Lite – це безкоштовна програма для настільних комп'ютерів, яка дозволяє отримувати доступ до Zoho Mail одночасно з ПК чи ноутбука. Вона поєднує звичний інтерфейс веб-версії з покращеною швидкістю та зручністю. TechExpert пропонує повний спектр послуг з налаштування, інтеграції та доопрацювання рішень на базі Microsoft Dynamics 365. Спеціалісти мають досвід роботи зі спільними галузями та бізнес-процесами, що дозволяє адаптувати систему під конкретні потреби замовника.

Аналіз існуючого стану технічної підтримки клієнтів організацій показав, що впровадження автоматизованої системи є бажаним етапом підвищення ефективності обслуговування клієнтів. Популярні інструменти, такі як Zendesk, Freshdesk і Zoho Desk, володіють широким функціоналом, але їхня вартість і надійність налаштування залишаються значним бар'єром для більшості

організацій. Використання Windows Forms для створення інформаційної системи техпідтримки дозволяє запропонувати альтернативні рішення, орієнтовані на локальні потреби організацій, із меншими витратами на впровадження та можливість гнучкої адаптації.

Нами розроблену концептуальну модель системи техпідтримки клієнтів. Розглянуто зв'язки між об'єктами бази даних. Користувач (user) створює заявку (ticket). Заявка (ticket) може включати кілька процесів (process), що були активні на комп'ютері користувача під час створення заявки. Заявка (ticket) пов'язана з категорією або типом (type). Ці зв'язки забезпечують можливість комплексного управління заявками та збору необхідної інформації для вирішення проблем користувачів.

На рисунку 2.1 представлено логічну модель бази даних, що відображає взаємозв'язки між об'єктами user, ticket, process і type. Ця модель забезпечує структуровану організацію даних, необхідних для ефективної роботи системи технічної підтримки.

Діаграма варіантів використання для системи технічної підтримки клієнтів зображена на рисунку 2.2. Вона містить основні актори, варіанти використання та взаємозв'язки між ними. Детальний аналіз варіантів використання та їх графічне представлення є основою для подальшої розробки системи, зокрема для реалізації інтерфейсу та моделювання бази даних. Це забезпечує ефективне планування, розробку та тестування всіх компонентів інформаційної системи технічної підтримки клієнтів.

Використання нотацій IDEF0 та IDEF3 у процесі моделювання інформаційної системи забезпечує забезпечило визначення чіткої структури функцій і послідовності дій, що полегшує розробку програмного забезпечення, забезпечуючи зрозумілу документацію для команди розробників. Створені діаграми у нотаціях IDEF0 та IDEF3 є невід'ємною частиною процесу розробки системи технічної підтримки клієнтів, сприяючи структурованому підходу до моделювання та оптимізації бізнес-процесів.

Інтеграція розроблених модулів з інтерфейсом користувача дозволяє створити цілісну систему, яка відповідає потребам усіх сторінок сайту. Усі компоненти, включаючи обробку заявок, управління користувачами та категоріями заявок, об'єднані в єдиний інтерфейс, що забезпечує ефективну взаємодію між користувачами та системою.

Для успішного проектування інформаційної системи технічної підтримки клієнтам необхідно вибрати відповідні інструменти, які забезпечують ефективну реалізацію як основних модулів, так і зручного інтерфейсу користувача. Таким чином, для реалізації інформаційної системи використовуються поєднання C#, SQL Server, Windows Forms та Visual Studio, що забезпечує ефективність розробки, інтеграцію між модулями та стабільність системи. Вибрані засоби можуть створити продуктивну інформаційну систему, яка відповідає сучасним вимогам до програмного забезпечення та забезпечує високу якість обслуговування клієнтів.

Нами виконано проектування основних модулів інформаційної системи. Нами написано код та створено модуль прийому заявок є базовим компонентом інформаційної системи технічної підтримки. Він забезпечує ефективний прийом та збереження даних, зменшуючи ризик втрати інформації та оптимізуючи подальшу обробку заявок.

Нами написано код та створено модуль обробки заявок є невід'ємною частиною системи, що дозволяє інженерам ефективно працювати із заявками клієнтів. Завдяки інтеграції з базою даних і зручному інтерфейсу цей модуль забезпечує прозорість процесів і підвищену ефективність технічної підтримки.

Нами написано код та створено модуль управління користувачами дозволяє адміністраторам ефективно керувати системами користувачів, підтримуючи її безпеку та прозорість. Завдяки функції автентифікації та журналу дій забезпечується високий рівень захисту інформації та контроль доступу до системи.

Розробку форм для інформаційної системи технічної підтримки клієнтів виконано за допомогою Windows Forms у середовищі розробки Microsoft Visual

Studio. Для цього створюється проект «Windows Forms Application (.NET Framework)», виконується необхідне налаштування, після чого отримуємо середовище для проектування інтерфейсу користувача програми (рис. 3.10).

Розроблений інтерфейс Windows Forms забезпечує ефективну взаємодію між клієнтами, інженерами та адміністраторами, що сприяє високій продуктивності системи.

Нами розроблено заходи із охорони праці під час розроблення та монтажу інформаційної системи технічної підтримки клієнтів.

Результати визначення показників економічної ефективності показують значне зниження витрат на обробку заявок, що забезпечує економію 20000 грн на місяць. Зменшення кількості працівників від 10 до 6 дозволяє скоротити витрати на оплату праці, а скорочення часу обробки заявки на 30 хвилин продуктивності системи. Впровадження інформаційної системи дозволяє не тільки зменшити витрати, але й оптимізувати процеси технічної підтримки, забезпечуючи кращий сервіс для клієнтів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бабак О.В. Розробка інформаційної технології формування запитів техпідтримки. Кваліфікаційна магістральна робота. СумДУ, 2018. 52 с.
2. Вишня В. Б. Основи інформаційної безпеки. Навчальний посібник / В. Б. Вишня, О. С. Гавриш, Е. В. Рижков. – Дніпро, 2020. – 128 с.
3. Вступ в С# [Електронний ресурс] – Режим доступу до ресурсу: <https://programm.top/uk/c-sharp/tutorial/introduction>
4. Глоба Л.С. Розробка інформаційних ресурсів та систем. Том 2. К.: ДУТ, 2018. 300 с.
5. Головка Г. В. – Методичні рекомендації до дисципліни «Технології захисту інформації».
6. Графічний редактор “Canva” [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://nus.org.ua/news/grafichnyj-redaktor-canva-stav-rovnistyuu-bezplatnym-dlya-osvityan>
7. Грищенко І.В. Розробка підсистеми CRM для спрощення ведення обліку замовлень та клієнтів. Кваліфікаційна робота. КНЕУ, 2019. 55 с.
8. Дяченко М.П. Основи розробки додатків з використанням Windows Forms. К.: МАУП, 2019. 21 с.
9. Жидецький В.Ц., Джигирей В.С., Мельников О.В. Основи охорони праці. Підручник. Вид. 5-е, доповнене. Львів: Афіша, 2012. 350с.
10. Засоби та методи захисту інформації [Електронний ресурс] – Режим доступу до ресурсу: <https://buklib.net/books/28625>
11. Захист інформації у сучасному світі [Електронний ресурс] – Режим доступу до ресурсу: <https://remonline.ua/blog/protection-of-information-in-modern-world>
12. Інформаційна та кібербезпека: соціотехнічний аспект. Київ: ДУТ, 2015. – 288 с.
13. Круглик О.В. Концепція організації інформаційної системи технічної підтримки. Вісник ХДУ, 2011. 12 с.

14. Лабунська Я.О. Розробка автоматизованої системи інформаційної підтримки клієнтів. Магістерська робота. СумДУ, 2020. 60 с.
15. Лехман С.Д., Рублев В.І., Рябцев Б.І. Запобігання аварійності і травматизму у сільському господарстві. К.: Урожай, 1993. 267 с.
16. Ложніков А.О. Побудова інформаційної системи ІТ-компанії «Прогрес» для підтримки клієнтів. Кваліфікаційна робота бакалавра. ТНТУ, 2024. 68 с.
17. Лунькін О.В. Система технічної підтримки клієнтів за допомогою рішень з відкритим кодом. Дипломна робота. СНУ ім. В. Даля, 2019. 45 с.
18. Правила забезпечення захисту інформації в інформаційних, телекомунікаційних та інформаційно-телекомунікаційних системах [Електронний ресурс] – Режим доступу до ресурсу: <https://www.kmu.gov.ua/npas/32791826>
19. Пфо О.М. Основні проблеми теорії захисту інформації / Пфо О.М. // Всеукраїнська науково-практична конференція 23-25 листопада 2016 року / Пфо О.М.. – Кропивницький, 2016.
20. Tryhuba A., Kondysiuk I., Tryhuba I., Koval N., Boiarchuk O., Tatomyr A. Intellectual information system for formation of portfolio projects of motor transport enterprises, in: I Workshop Information Technologies in Energy and Agro-industrial Complex, ITEA-WS 2021, CEUR Workshop Proceedings 3109, Dubliany, Lviv region, 2021, pp. 44–52.
21. Технології захисту інформації. // Проблеми захисту інформації у сучасних ІС, 2015.
22. Тригуба А.М., Кондисюк І.В., Татомир А.В., Шолудько Я.В., Боярчук О.В. Інтелектуальна інформаційна система формування портфелів проектів автотранспортних підприємств. Інформаційні технології в енергетиці та агропромисловому комплексі: матеріали X-ї міжнародної наукової конференції, присвяченої 165-річчю університету. Львів-Дубляни, 2021, С. 113–115.

23. Що таке специфікація вимог: визначення, найкращі інструменти та методи [Електронний ресурс] – Режим доступу до ресурсу: <https://visuresolutions.com/uk/blog/requirements-specification>.

24. Computer Telephony Integration Framework. URL: <https://partnersupport.freshworks.com/support/solutions/articles/221418-computer-telephony-integration-framework>

25. Desktop Guide (Windows Forms .NET) [Електронний ресурс]. 2023. – Режим доступу до ресурсу: <https://learn.microsoft.com/uk-ua/dotnet/desktop/winforms/overview/?view=netdesktop-7.0>

26. Introduction to the Support agent interface (standard agent interface). URL: <https://support.zendesk.com/hc/en-us/articles/4408883355546-Introduction-to-the-Support-agent-interface-standard-agent-interface>

27. Microsoft Dynamics 365: послуги з впровадження та доопрацювання CRM та ERP рішень. URL: <https://techexpert.ua/it-services/md-365-services-for-implementation-and-refinement-of-crm-and-erp-solutions/>

28. Microsoft Visual Studio Professional 2022 [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://www.softkey.ua/catalog/programming/microsoft-visual-studio-professional-2019-olp>

29. Test Case [Електронний ресурс] – Режим доступу до ресурсу: <https://qalight.ua/ru/baza-znaniy/test-case>

30. ZOHO Partner. URL: <https://uk.crmoz.com/blogs/post/how-to-download-and-install-zoho-for-windows-mac-or-linux>

Додатки

Додаток А

Фрагмент коду для інтерфейсу клієнта

```
using System;
using System.Windows.Forms;

namespace ClientInterface
{
    public partial class ClientForm : Form
    {
        public ClientForm()
        {
            InitializeComponent();
        }

        private void InitializeComponent()
        {
            this.Text = "Інтерфейс клієнта для створення заявки";
            this.Size = new System.Drawing.Size(800, 600);

            // Поле для заголовка заявки
            TextBox titleTextBox = new TextBox
            {
                Location = new System.Drawing.Point(100, 50),
                Size = new System.Drawing.Size(600, 30),
                PlaceholderText = "Введіть заголовок заявки"
            };

            // Поле для опису заявки
            TextBox descriptionTextBox = new TextBox
            {
                Location = new System.Drawing.Point(100, 100),
                Size = new System.Drawing.Size(600, 150),
                Multiline = true,
                PlaceholderText = "Введіть опис проблеми"
            };

            // Кнопка для додавання файлів
            Button addFileButton = new Button
            {
                Text = "Додати файли",
                Location = new System.Drawing.Point(100, 300),
                Size = new System.Drawing.Size(150, 30)
            };
            addFileButton.Click += (s, e) => MessageBox.Show("Додати файл");

            // Кнопка для надсилання заявки
            Button submitButton = new Button
```

```
{
    Text = "Надіслати заявку",
    Location = new System.Drawing.Point(550, 300),
    Size = new System.Drawing.Size(150, 30)
};
submitButton.Click += (s, e) => MessageBox.Show("Заявка успішно створена!");

// Додавання елементів на форму
this.Controls.Add(titleTextBox);
this.Controls.Add(descriptionTextBox);
this.Controls.Add(addFileButton);
this.Controls.Add(submitButton);
}

[STAThread]
static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Application.Run(new ClientForm());
}
}
```