

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра Інформаційних систем
Освітній ступінь Бакалавр
Спеціальність 122 Комп'ютерні науки
(код і назва)
Освітньо-професійна програма Комп'ютерні науки
(назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри Чумаченко С.М

“ ” _____ 20__ року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Кішки Сергія Олександровича

(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення інформаційної підсистеми відділу збуту для кондитерського дому «Вацак»

керівник роботи Бобрівник Катерина Євгенівна. к.т.н.,

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

затверджені наказом закладу вищої освіти від “09” 11.2020 року № 932-КС

2. Строк подання здобувачем роботи 26.01.2021

3. Вихідні дані до роботи

1. Список продукції, перелік дистриб'юторів

2. Правила і норми правила, статут підприємства

3. Інформація з офіційної Інтернет сторінки підприємства

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

1. Системний аналіз діяльності підприємства та постановка задачі

2. Технічне завдання

3. Розробка комплексу задач автоматизації

4. Охорона праці

5. Перелік графічного матеріалу

1. Функціональна модель.

2. Логічна модель

3. Фізична модель

4. Діаграма бази даних

5. Скріншоти інтерфейсу програмного додатку _____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ 1	Бобрівник К. Є.		
Розділ 2	Бобрівник К. Є.		
Розділ 3	Бобрівник К. Є.		
Розділ 4	Бобрівник К. Є.		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Системний аналіз діяльності підприємства		Виконано
2	Постановка задачі на розроблення підсистеми		Виконано
3	Розроблення функціональної моделі підприємства		Виконано
4	Проектування бази даних		Виконано
5	Створення додатку користувача		Виконано
6	Написання інструкції користувача		Виконано
7	Оформлення пояснювальної записки		Виконано
8	Оформлення презентації		Виконано

Здобувач _____

(підпис)

_____ (прізвище та ініціали)

Керівник роботи _____

(підпис)

_____ (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота «Розроблення системи інформаційної підтримки відділу збуту Кондитерського Дому «ВАЦАК», розроблено Кішка С. О.

Робота складається з 90 сторінок, 9 таблиць, 14 рисунків, 5 додатків та 17 літературних джерел.

Було досліджено, використовуючи методи моделювання, системного аналізу для розробки системи інформаційної підтримки діяльності відділу збуту на підприємстві Кондитерський Дім «ВАЦАК». Було представлено декілька систем аналогів, а також проведення їх порівняння.

Розроблена функціональна модель роботи підсистеми, логічна та фізична моделі, за допомогою CASE-засобу AllFusion ERWin. Інтерфейс підсистеми було зроблено за допомогою Microsoft Visual Studio 2015.

У кваліфікаційній роботі обґрунтовано доцільність створення підсистеми, розроблено функціональну модель, здійснення вибору ПЗ, розроблена інструкція для користувача. Описано заходи з охорони праці підприємства. Проведений розрахунок економічної ефективності від впровадження підсистеми.

Ключові слова: СИСТЕМА ІНФОРМАЦІЙНОЇ ПІДТРИМКИ, ВІДДІЛ ЗБУТУ, ФУНКЦІОНАЛЬНА МОДЕЛЬ, ЛОГІЧНА МОДЕЛЬ, ФІЗИЧНА МОДЕЛЬ, ІНСТУКЦІЯ КОРИСТУВАЧА, MICROSOFT SQL 2016, MICROSOFT VISUAL STUDIO 2015.

SUMMARY

The diploma project "Development of information support systems for the sales department of the Confectionery House" VATSAK ", Kishka S . O.

The work consists of 90 pages, 9 table, 14 figures, 5 appendices and 17 references.

The diploma project explores the methods of modeling, system analysis for the development of information support systems for the sales department at the Confectionery House "VATSAK". Several system analogues were presented and compared.

In work with the public system of creation, the functional model is developed, the choice of the software is offered, the user's instruction is developed. Occupational safety measures are described. The calculation of economic efficiency from the implementation of systems at the enterprise.

Developed a functional model of the information system, as well as logical and physical models using CASE - using All Fusion BPWin and ERWin, respectively. The system interface was developed using Microsoft Visual Studio 2015.

Keywords: INFORMATION SUPPORT SYSTEM, SALES DEPARTMENT, FUNCTIONAL MODEL, LOGICAL MODEL, PHYSICAL MODEL, USER INSTRUCTION, USER 2016

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ДІЯЛЬНОСТІ ВІДДІЛУ ЗБУТУ КОНДИТЕРСЬКОГО ДОМУ «ВАЦАК» ТА ПОСТАНОВКА ЗАДАЧІ НА ПРОЕКТУВАННЯ.....	9
1.1 Опис та особливості функціонування Кондитерського Дому «ВАЦАК»	9
1.1.1 Загальна характеристика Кондитерського Дому «ВАЦАК»	10
1.1.2 Організаційна структура підприємства, роль та взаємодія підрозділів	12
1.2 Характеристика роботи відділу збуту	12
1.3. Розроблення функціональної моделі.....	13
1.3.1. CASE-засіб BPWin	13
1.3.2 Моделювання діяльності відділу збуту за допомогою CASE-засобу BPWin	16
1.4. Аналіз аналогів систем розробки	19
1.4.1 Німецька корпорація «SAP»	19
1.4.2 Система Microsoft Dynamics NAV	20
1.4.4 Порівняння систем аналогів.....	24
1.5 Постановка задачі на проектування	25
РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ	27
2.1. Загальні положення.....	27
2.2. Призначення і цілі створення підсистеми	27
2.2.1. Призначення підсистеми	27
2.2.2. Цілі створення підсистеми.	27
2.3. Характеристика об'єкта автоматизації.	28
2.4. Вимоги до підсистеми	28
2.5. Вимоги до видів забезпечення.	34
2.6. Склад і зміст робіт по створенню підсистеми.....	38

2.7. Порядок контролю і приймання підсистеми	38
2.8. Вимоги до складу і змісту робіт із підготовки до введення підсистеми в дію.....	39
2.9. Вимоги до документації.....	39
2.10. Джерела розробки	39
РОЗДІЛ 3. РОЗРОБКА КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ.....	41
3.1 Обґрунтування вибору програмного забезпечення використовуваного для розробки системи	41
3.1.1 Структура програмного забезпечення	41
3.1.2 Опис логічної та фізичної моделі бази даних розробки	43
3.2 Реалізація логічної і фізичної бази даних.....	45
3.3. Інформаційне забезпечення розробки Microsoft SQL 2016	46
3.4 Інструкція користувача системи.....	48
3.4.1 Початок роботи і головне меню системи	48
3.4.2 Форми введення і перегляду.....	48
3.4.3. Виконання пошуку.....	50
3.5 Економічна ефективність використання підсистеми	51
РОЗДІЛ 4. ОХОРОНА ПРАЦІ	59
ВИСНОВОК.....	66
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	67
ДОДАТКИ.....	68

ВСТУП

У будь-якій організації поступово накопичується пристойний обсяг даних, частина з яких потрібна лише зрідка, відправити в архів їх не можна, і більше часу вони просто займають ресурси. Джерела даних сьогодні найрізноманітніші, і організаціям доводиться мати справу з декількома типами даних, аналіз яких стає проблемою.

В даній дипломній роботі розглядається підприємство Кондитерський Дім «ВАЦАК», а саме його відділ збуту готової продукції. Підприємство надає послуги в сфері кондитерської промисловості, забезпечуючи споживачів свіжою випічкою.

Однією із головних задач підприємства на даний час повинно бути зосередження на торгівлі, так як це призводить до постійного забезпечення каналами збуту, за рахунок цього досягається ефективність виробництва і витісняє конкурентів, які у БКК досить серйозні, завдяки цьому конкуренти змогли досягти зменшення цін на 10-30%. [15]

На підприємстві діє однорівневий канал збуту – продукція переходить від виробника через організацій роздрібною торгівлі до споживачів.

Метою даної бакалаврської роботи є створення інформаційної системи підтримки відділу збуту підприємства Кондитерський Дім «ВАЦАК». Та її впровадження на виробництві.

Зараз підприємство потребує вирішення завдань системи входить:

- облік менеджерів;
- облік готової продукції;
- активні договори;
- виконання пошуку;

РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ДІЯЛЬНОСТІ ВІДДІЛУ ЗБУТУ КОНДИТЕРСЬКОГО ДОМУ «ВАЦАК» ТА ПОСТАНОВКА ЗАДАЧІ НА ПРОЕКТУВАННЯ

1.1 Опис та особливості функціонування Кондитерського Дому «ВАЦАК»

Кондитерський Дім «ВАЦАК» цього року відзначає своє 19 річчя. Самовіддача, впровадження бізнес-ідей, сучасних технологій, створення гідних умов праці – такі принципи роботи були покладені у стратегію бізнесу з першого дня заснування. За цей час підприємство встигло не тільки міцно стати на ноги, а й розширити спектр продукції та географію своєї виробничої діяльності, розгорнути по всій території України мережу фірмових магазинів, збільшити кількість ділових партнерів, що дозволило зайняти почесне місце в кондитерській галузі та отримати популярність серед споживачів. Кондитерський Дім «Вацак» має амбітні плани на майбутнє. За ці 19 років сталося багато цікавих подій і ми вдячні, що ви були з нами. Ми йдемо вперед незважаючи на перешкоди. Ми готові вирішувати найважчі завдання, не зупинятися на досягнутому та підкорювати нові вершини. [2]

Асортимент може задовольнити практично будь-якого клієнта, з будь-якими смаками та потребами. До того ж, наявна виробнича база з сучасними імпортними технологічними лініями та кваліфікованими кондитерами дає змогу втілювати всі забаганки. Наше фірмове кредо – якість продукції та натуральна сировина. [2]

У Кондитерському Домі «Вацак» є своя тематична галерея: весільні торти, виконані у витонченій манері, торти для різних святкових заходів з нанесенням малюнка, корпоративні торти з фірмовим логотипом чи брендом, торти на дитячу тематику, з різними кольоровими, їстівними фігурками чи казковими персонажами. Кожен такий десерт – це витвір кулінарного мистецтва, в які наші майстри вкладають всю свою душу та талант. Ми

творимо солодощі, які дають відчуття рідного, щастя та домашнього затишку[2].

1.1.1 Загальна характеристика Кондитерського Дому «ВАЦАК»

КД «Вацак» виготовляє понад 150 найменувань цукерок, шоколаду, печива, зефіру, мармеладу й тортів. Також виробник виконує індивідуальні замовлення на торти. Переважна більшість продукції (70 із 150-ти) — саме торти: вагові, у коробках та на замовлення[3].

Таблиця 1.1. Загальні відомості БКК

Реквізити	Характеристики
Повна назва	Кондитерський Дім "Вацак"
Юридична та поштова адреса	Україна, Вінницька область, м. Могилів-Подільський вул. Вірменська, 4
Підприємство зареєстровано	2002 року
Форма власності	колективна
Галузь	харчова
Види діяльності	виробництво борошняних кондитерських виробів, тортів і тістечок нетривалого зберігання

Асортимент компанії виготовляє продукцію на основі традиційних та власних рецептів, що відповідають вимогам ДСТУ. На відміну від багатьох конкурентів, продукцію Кондитерського Дому виготовляють на основі натурального вершкового масла, без використання шкідливих компонентів. Торти декорують винятково вручну їстівними фігурками чи казковими

персонажами, тому кожен десерт є справжнім витвором кондитерського мистецтва.

Управлінням поточною діяльністю підприємства займаються п'ять чоловік на чолі з головою правління, який також є генеральним директором підприємства:

Директору підпорядковуються:

- Фінансовий директор підприємства
- Головний бухгалтер підприємства
- Технічний директор підприємства
- Директор з персоналу
- Директор по виробництву
- Директор з логістики

До складу організаційної структури входять такі відділи:

- Юридичний відділ
- Бухгалтерія
- Відділ головного технолога
- Господарський відділ
- Відділ поставок
- Відділ з охорони праці
- Економічний відділ
- Відділ головного механіка

Фінансовому директору підприємства підпорядковуються такі підрозділи:

- Бухгалтерія
- Економічний відділ

Головному бухгалтеру підкоряються:

- Бухгалтер
- Архів

1.1.2 Організаційна структура підприємства, роль та взаємодія підрозділів

Організаційна структура Кондитерського дому «ВАЦАК» носить яскраво виражений функціональний характер. Характерною рисою є угруповання персоналу за окремими напрямками діяльності, таким як, виробництво, збут, фінанси, правові питання і т.д. Відповідно до цих напрямів формується директорат і структурні підрозділи підприємства. При формуванні організаційної структури був використаний принцип функціональної департаменталізації, тобто поділу організації на окремі елементи, кожний з яких має свою чітко визначену конкретну задачу і обов'язки.

1.2 Характеристика роботи відділу збуту

Організаційна структура компанії спрямована на отримання кінцевого результату метою якого є створення організаційної структури полягає в розподілі обов'язків і координації діяльності членів групи, щоб при виконанні поставлених завдань вони діяли як одна команда. Саме тому створення структури збуту, виконує завдання відповідно свого підрозділу та формується виходячи з маркетингових цілей компанії.

При створенні організаційної структури збуту враховано наступне:

- поділ праці і спеціалізація співробітників повинні бути вигідним для компанії;
- організація продажів повинна забезпечувати стабільність і безперервність продажів компанії;
- організація продажів повинна забезпечувати координацію різних видів збутової діяльності, виконуваних окремими співробітниками або підрозділами фірми.

Поділ і спеціалізація праці співробітників спрямовані на досягнення цілей організації. Чим більше організаційних завдань вирішено різними фахівцями, тим більший прибуток торимає компанія завдяки кваліфікованим кадрам але в свою чергу складніше координувати їх роботу. Збут здійснюється силами зовнішніх агентів або посередників, труднощі зростають, оскільки менеджер не може безпосередньо керувати їхньою роботою і далеко не завжди може контролювати їх дії.

Координація та інтеграція діяльності співробітників відділу продажів організації будується на наступних принципах:

- орієнтація на потреби і інтереси клієнтів компанії;
- взаємодія з іншими підрозділами компанії (виробництвом, конструкторським бюро, логістикою, фінансовим відділом і ін.);
- узгодженість завдань між спеціалізованими групами, які виконують різні функції, в рамках одного відділу продажів.

Створюючи власний відділ продажів вигідніше ніж передавати збут в аутсорсинг, але формується з урахуванням багатьох факторів, але до числа значущих для Кондитерського Дому «ВАЦАК» віднесено наступні чотири:

- економічна доцільність;
- необхідність контролю і координації;
- транзакційні витрати;
- стратегічна гнучкість.

Тому є необхідність розробки інформаційної системи підтримки діяльності відділу збуту.

1.3. Розроблення функціональної моделі

1.3.1. CASE-засіб BPWin

Світовий лідер в області CASE-технології пропонує потужний засіб системного аналізу ділової та виробничої активності, що дозволяє відстежувати відповідність структури бізнесу, документообігу, фінансових потоків жорстким і динамічним вимогам сучасної економіки.

Система BPwin допоможе підвищити конкурентоспроможність, оптимізувати процеси управління.

BPwin - це незамінний інструмент менеджерів і бізнес-аналітиків, а починаючи з версії 1.8, в яку включена підтримка діаграм потоків даних і методики IDEF3 (BPwin Professional), стає в руках системних аналітиків і розробників і потужним засобом моделювання процесів при створенні корпоративних інформаційних систем.

BPwin має інтуїтивно-зрозумілим графічним інтерфейсом, допомагає швидко створювати і аналізувати моделі з метою оптимізації ділових і виробничих процесів. Застосування універсальної графічної мови бізнес-моделювання IDEF0 забезпечує логічну цілісність і повноту опису, необхідну для досягнення точних і несуперечливих результатів. За допомогою набору графічних інструментів BPwin дозволяє легко побудувати схему процесу, на якій показані вихідні дані, результати операцій, ресурси, необхідні для їх виконання, що управляють, взаємні зв'язки між окремими роботами.

BPwin може генерувати звіти безпосередньо в форматі MS Excel і Word для подальшої обробки та використання в інших додатках. Зв'язок з ERwin (моделювання даних в стандарті IDEF1X) дозволяє скоротити час проектування та розробки складних інформаційних систем. Для системних аналітиків тісна інтеграція BPwin з інструментом проектування баз даних відкриває унікальні можливості по створенню комплексних систем, в яких ERwin служить для опису інформаційних об'єктів системи, в той час як BPwin відображає функціональні особливості предметної області. Пов'язуючи сутності й атрибути моделі даних з інформацією про виконувани дії, Ви

можете продовжити аналіз процесів на новому рівні з одночасною перехресної перевіркою моделей процесів і даних.

Основні характеристики VPwin:

- розвинена методологія функціонального моделювання на основі IDEF0;
- потужні редактори для опису операцій, зв'язків і обчислення витрат на виконання робіт;
- ієрархічна структура діаграм, що полегшує послідовне уточнення елементів моделі;
- контекстні діаграми для опису меж системи, області дії, призначення об'єктів;
- декомпозиційні діаграми для опису особливостей взаємодії різних процесів;
- розширені можливості по підтримці посилальної цілісності;
- підтримка методології IDEF3;
- експорт моделей в засоби імітаційного моделювання;
- інтеграція і зв'язок із засобом проектування баз даних erwin (методологія idef1x);
- підтримка властивостей, які визначаються користувачем. опис моделей може бути розширено за рахунок властивостей, що визначаються користувачем, включаючи мультимедійні документи;
- інтеграція з modelmart, що підтримує потужний набір інструментальних програмних засобів, що забезпечують спільне (групове) проектування і розробку програмних систем, включаючи механізми об'єднання моделей і аналізу змін, контроль версій, можливість створення "компонент" моделі і т.д. для організації сховища моделей в modelmart використовуються субд на платформах

oracle, sybase, informix або sql server. крім того, підтримуються прямі зв'язки modelmart з erwin і bpwin;

- зручний інтерфейс користувача. у розпорядженні користувачів є провідник, який став звичним в середовищі windows 95 / nt, що дозволяє легко переходити з однієї діаграми на іншу простим переміщенням по "дереву" провідника;
- розширена архітектура. bpwin підтримує 16- та 32-х розрядні системи, дозволяючи організувати спільну роботу для всіх учасників проекту;
- автоматична підтримка зміни розмірів. bpwin підтримує автоматичну настройку розмірів діаграм і можливість зміни масштабів зображення моделей.

1.3.2 Моделювання діяльності відділу збуту за допомогою CASE-засобу BPWin

Модель роботи відділу збуту повинна забезпечити оперативне отримання повної і достовірної інформації від усіх ділянок збуту, пов'язаних між собою в єдину мережну структуру. Тому, з'являється потреба розробляти моделі на новому рівні, використовуючи нові засоби розробки інформаційних систем, забезпечивши виконання обов'язкових умов: повноти, своєчасності і достовірності інформації. Межі моделі, від отримання плану випуску продукції до відвантаження продукції замовнику[4].

Потрібно розробити функціональну модель діяльності відділу збуту в середовищі BPwin, створюємо логічну модель БД в середовищі ERwin[4].

Зручною мовою моделювання бізнес-процесів є IDEFO. Методологія IDEFO описує побудову ієрархічної системи діаграм – одиничних описів фрагментів системи. Під моделлю в IDEFO розуміють опис системи (текстовий і графічний), який повинен дати відповідь на деякі раніше

визначені запитання. В IDEF0 система є як сукупність взаємодіючих робіт чи функцій. Така чисто функціональна орієнтація є принциповою – функції системи аналізуються незалежно від об'єктів, якими вони оперують. Це дозволяє більш чітко змодельовати логіку і взаємодію процесів організації. Взаємодію з навколишнім середовищем описується як вхід (те, що перероблюється системою), вихід (результат діяльності системи), управління (стратегії і процедури, під управлінням яких виконується робота) і механізми (ресурси, необхідні для проведення роботи). Знаходячись під управлінням, система перетворює входи у виходи, використовуючи механізми. Процес моделювання будь-якої системи в IDEF0 починається з визначення контексту, тобто абстрактного рівня опису системи в цілому. В контекст входить визначення суб'єкта моделювання, цілі і точки зору на модель. Під суб'єктом розуміється сама система, при цьому необхідно визначити, що ми будемо розглядати як компоненти системи, і як зовнішні діяння. Спочатку необхідно визначити область моделювання. Вона визначає напрям моделювання і коли повинна бути закінчена модель. Проведемо розробку моделі бізнес-процесу, застосувавши функціональне моделювання бізнес-процесу з використанням нотації IDEF0 і CASE-средства для автоматизованого проектування All Function Process Modeler[4].

На рисунку 1.2 представлена діаграма процесу збуту з використанням інформаційної системи, яка відображає загальну картину автоматизації процесу. На рисунку 1.3 представлена деталізація діаграми, отримана методом декомпозиції.

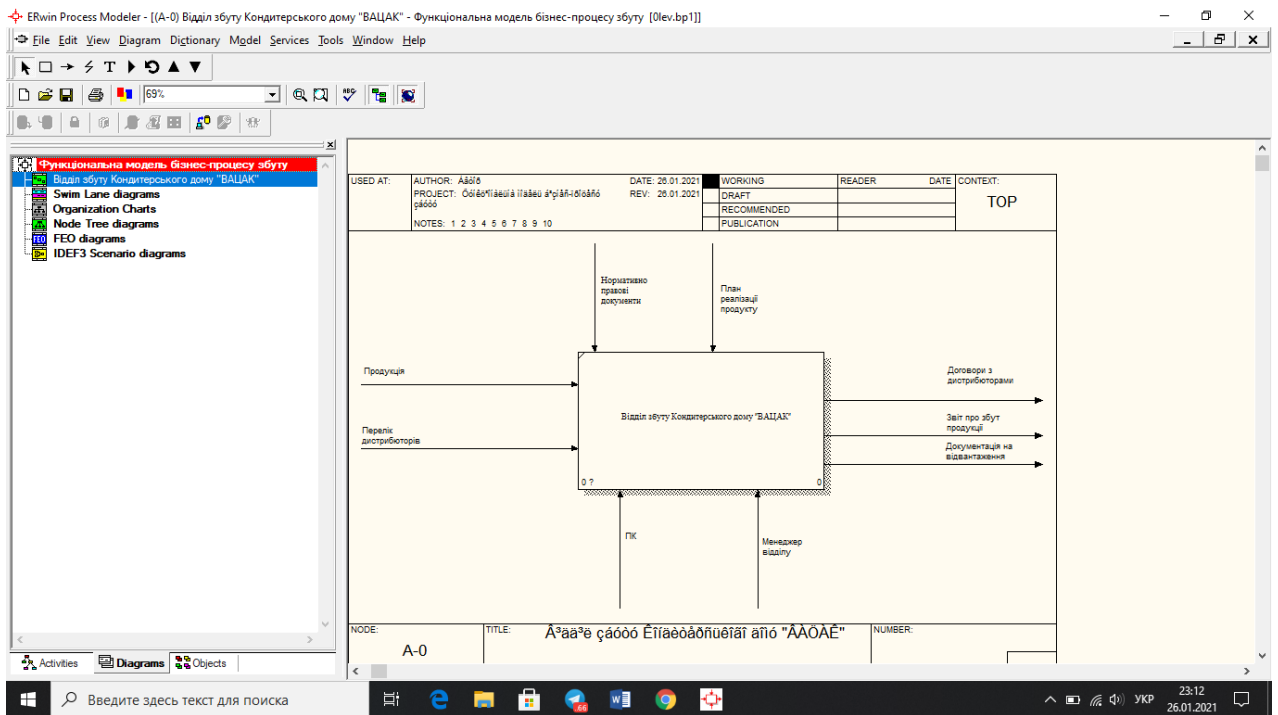


Рис. 1.2. Функціональна модель бізнес-процесу збуту

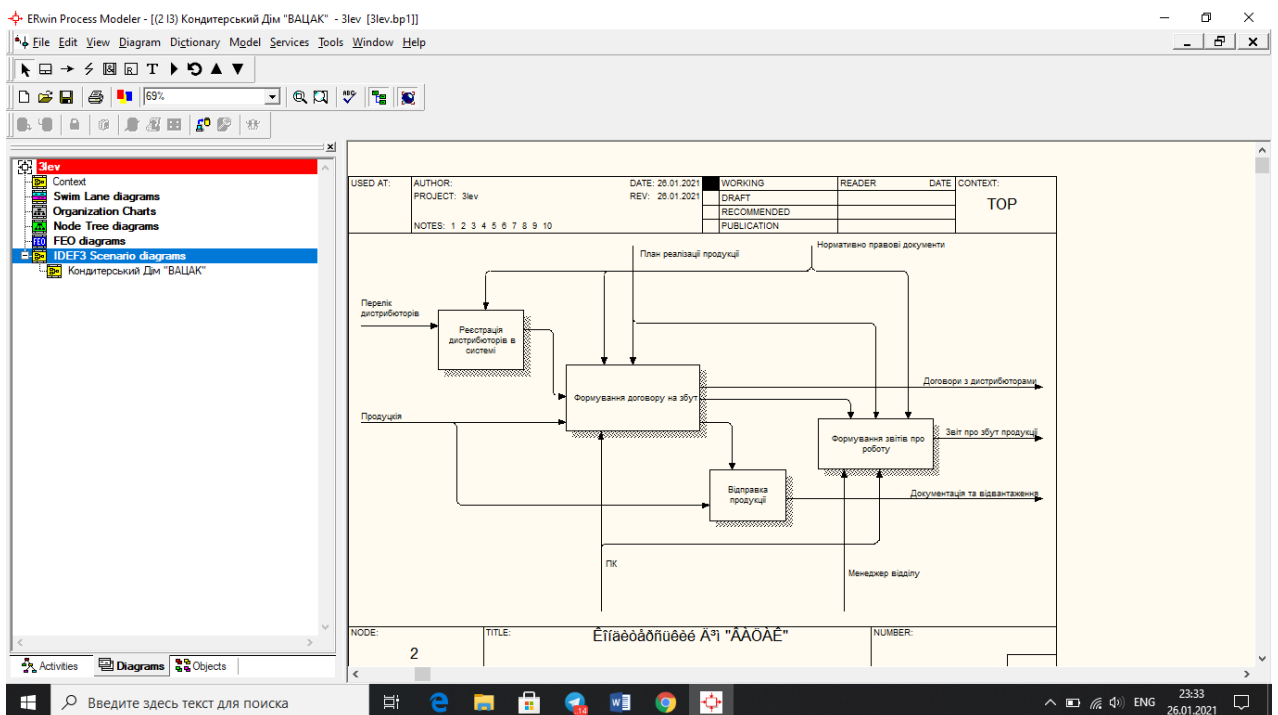


Рис. 1.3. Декомпозиція моделі бізнес-процесу збуту

Як видно з малюнка 1.3, процес організації збуту на підприємстві можна розділити на три підпроцеса:

1. На етапі формування угоди на збут проводиться введення даних про домовленості з замовниками та всі нормативно правові акти та плани реалізації.

2. На етапі формування графіку збуту фіксуються угоди, продукція для реалізації, графіки поставок і інша інформація.

3. На третьому етапі фіксується план випуску продукції, кількість відвантаженої продукції та враховуються всі супровідні документи.

1.4. Аналіз аналогів систем розробки

1.4.1 Німецька корпорація «SAP»

SAP - це автоматизована система, яка пропонує комплекс рішень для вибудовування спільного інформаційного простору на базі підприємства і ефективного планування ресурсів і робочих процесів.

Її інструменти можна використовувати як окремо, так і комбінуючи. Головне, щоб середовище було єдиної - це дозволить забезпечити максимальну ефективність проведення та актуалізації даних між різними відділами або іншими функціональними одиницями компанії.

Модель ПО включає в себе 3 ланки:

- клієнт,
- сервер,
- СУБД.

Така структура дозволяє кінцевому споживачеві з зручністю об'єднувати ключові рішення в двох таких сферах як:

- бухоблік і складання звітності - можна фіксувати всі виробничі витрати, управляти коштами і замовленнями, відзначати інші важливі результати;
- логістика - з плануванням, збутом і стратегічним управлінням відразу, включаючи оформлення рахунків, здійснення прямих

продажів і регулярні відвантаження товарів; сюди ж входить і постійне матеріально-технічне постачання, з проведенням закупівель і контролем запасів.

У загальному випадку ПО прискорює процеси обробки даних і дозволяє оптимізувати курс розвитку використовує його компанії, але буде ефективним тільки при логічною структурі організації і при наявності гарної взаємозв'язку між різними її підрозділами.

Це цілий комплекс рішень, що володіє наступними функціями:

- автоматизація праці бухгалтера;
- спрощення торговельних і складських операцій;
- полегшення обліку кадрів, фінансів, акцій та інших активів;
- модернізація логістики;
- складання максимально наочних зарплатних графіків.

Це ПО вельми функціональне, включає широке розмаїття модулів (важливою частиною є додатково підлаштовуватися), і дозволяє вирішувати різноманіття завдань. Воно порівняно дороге, але виправдовує себе в довгостроковій перспективі постійної роботи великого підприємства.

1.4.2 Система Microsoft Dynamics NAV

Корпорація Парус - один з російських розробників інформаційних систем для державного управління і бізнесу. У вітчизняній ІТ-галузі компанія працює з 1990 року.

Ми створюємо рішення ERP-класу: з їх допомогою можна автоматизувати будь-які процеси в області фінансово-господарської діяльності, управління НДДКР і виробництвом.

20 регіональних відділень і 200 партнерів в містах Росії впроваджують і супроводжують програмні продукти «Парус» відповідно до єдиних стандартів корпорації.

Додатки «Парус» використовують більше 25 тисяч організацій: федеральні відомства, регіональні адміністрації, органи місцевого самоврядування, державні та муніципальні установи, великі територіально-розподілені корпорації, провідні підприємства оборонно-промислового комплексу, машинобудівні заводи, науково-дослідні інститути.

Нашим клієнтам важливі високі показники продуктивності і масштабованості, інтеграційна гнучкість, широкий функціонал і можливість необмеженого розвитку системи. Це забезпечує замовникам високу ефективність IT-інвестицій, зниження трудовитрат, підвищення якості даних для планування, контролю та аналізу.

Navision - конкурентні переваги. Великі можливості для бізнесу.

Комплексна система управління підприємством Microsoft Dynamics NAV представляє в даний час результат багаторічної праці величезного числа професіоналів в області автоматизації. Витоки цієї системи йдуть від розробленої з 1983 року системи Navision. У ті роки створенням, розвитком, впровадженням та професійної підтримкою цього програмного продукту займалася створена в Данії Йеспер Балсером, Пітером Бангом і Торбеном Виндом однойменна компанія Navision. Спочатку система була створена як система автоматизації бухгалтерського обліку для Данії і Норвегії, а починаючи з 90-х років 20-го століття географія її застосування розширюється на Німеччину, Іспанію, Австрію. З цього часу Navision стає визнаною системою в Європі, а з 1992 року програма виходить на ринок США. Широкі функціональні можливості, простота у використанні, надійність роботи сприяли швидкому зростанню популярності Navision. При цьому функціональність системи постійно розширюється і виходить за рамки фінансового контуру - система розвивається в бік комплексної автоматизації

підприємств. У 2002 році компанія Navision була придбана корпорацією Microsoft і для подальшого розвитку системи було створено окремий підрозділ Microsoft Business Solutions. З цього періоду відбувається якісний стрибок у розвитку системи, вона стає однією з найпопулярніших у всьому світі. У 2004 році Microsoft починає ребрендинг продукту від Navision до Microsoft Dynamics NAV. Саме під цим найменуванням він і присутній на ринку бізнес-систем зараз.

Navision першим з комплексних ERP-систем світового класу був спеціально адаптований під особливості Росії і схвалений Міністерством Фінансів Російської Федерації. Зараз в Росії знаходиться окремий підрозділ Microsoft з підтримки системи в рамках відповідності вимогам законів.

З найперших версій в Navision був закладений принцип матричної настройки бізнес-процесів, що обумовлює простоту і логічність його налаштування. Формування фінансових проводок відбувається за принципом "від бізнес-процесу". Налаштовані в системі бізнес-процеси дозволяють повністю виключити некоректність формування облікових даних.

Navision має пред-налаштовану типову конфігурацію, за рахунок чого істотно скорочуються терміни на її впровадження. Ця типова конфігурація увібрала в себе особливості кращих рішень, побудованих за всю історію впроваджень Navision. Використовуючи типову конфігурацію продукту, клієнти мають можливість оптимізувати свої процеси, застосувавши на практиці багаторічний досвід споріднених компаній.

Система характеризується її вираженою спрямованістю на автоматизацію фінансового обліку компанії як ядра ERP. Це було характерною рисою перших версій Navision і залишається ключовим принципом Microsoft Dynamics NAV зараз.

Цінова доступність більшості підприємств малого і середнього бізнесу. В системі ліцензуються конкурентні (одночасно працюють) користувачі.

Microsoft Dynamics NAV розвивається з урахуванням останніх технологічних досягнень в області інформаційних технологій. Так в даний час широко застосовуються можливості web-сервісів. На практиці це означає, що доступ до даних Navision може бути отриманий з будь-якого комп'ютера, підключеного до інтернет-мережі з використанням звичайного інтернет-браузера. Співробітники компаній отримують всі переваги мобільності.

MS Navision має надійну систему безпеки і в-зокрема, повністю відповідає вимогам щодо захисту персональних даних громадян РФ. Безпека інформації забезпечується як на рівні самої бази даних, так і на рівні настраюється рольового доступу всередині самого додатка.

Архітектура системи передбачає можливість вибору між використанням в якості СУБД власної бази даних Navision або бази даних MS SQL. Це визначає такий важливий параметр як масштабованість. Для невеликих компаній з малою кількістю співробітників цілком достатнім буде Navision-сервер баз даних, а для великих клієнтів більше підійде SQL-сервер від Microsoft. При цьому забезпечується можливість переходу з однієї бази даних на іншу в процесі експлуатації системи без втрати даних.

Високі показники швидкодії системи, забезпечуються застосованою технологією динамічного індексного підсумовування. Всі найбільш важливі аналітичні дані користувачі отримують в режимі On-line.

Вбудована в систему середовище розробки C / SIDE. Navision спочатку проектувався як система з відкритим кодом, що дозволяє реалізовувати всі потрібні клієнтам нюанси. Ця обставина дозволила створити на базі платформи Navision велике число різних спеціалізованих рішень, що враховують особливості того чи іншого бізнесу. Роздрібна та оптова торгівля, виробничі та сервісні компанії, фінансовий і транспортний сектор - Navision успішно справляється з усіма цими специфічними завданнями.

Необмежена кількість ієрархічних аналітичних вимірів. Всі необхідні зрізи для аналізу бізнесу можуть бути без обмежень закладені в систему Navision і забезпечити команду управління необхідними даними.

Широкі технологічні можливості для інтеграції з іншими додатками дозволяють фахівцям реалізовувати навіть найскладніші технічні рішення по взаємодії Navision з багатьма програмними додатками.

Microsoft Navision - перевірена практикою система, що має більше 30 000 інсталяцій у світі. Досвід реальних впроваджень узагальнено та систематизовано у багатьох методичних матеріалах, наданих своїм клієнтам партнерами Microsoft.

Microsoft проводить регулярні навчальні семінари як для партнерів, так і для клієнтів Navision. Плани щодо подальшого розвитку системи поопублікуються в регулярно випускаються анонсах, що дозволяє компаніям своєчасно планувати стратегічні і тактичні кроки в сфері розвитку своєї ІТ-інфраструктури. Одним з останніх стратегічних рішень Microsoft стала концепція розвитку в системі Navision ролі-орієнтованого інтерфейсу, спрямованого на процесну уніфікацію роботи співробітників.

Для детального ознайомлення з можливостями системи клієнтам надається можливість завантажити демонстраційну версію продукту або пройти навчання у авторизованих партнерів Microsoft.

1.4.4 Порівняння систем аналогів

При порівнянні систем: Німецька корпорація «SAP» та «Парус», було визначено, що вони всі підходять для автоматизації роботи відділу збуту та підвищення його ефективності. Порівняння конкретних особливостей приведено у табл. 1.2

Таблиця 1.2. Результати порівняння систем аналогів

Система Особливості	Німецька корпорація «SAP»	Microsoft Dynamics NAV
Взаємодія з клієнтами	+	+
Розмір підприємства	Середнє – велике	Середнє – велике
Ціна за одну ліцензію	Від 7000-35000 грн	37 000 грн
Адаптація до підприємства	+	-
Формування документів	+	+
Маркетингові кампанії	+	+
Бухгалтерський облік	+	-
Інтеграція з Microsoft Office	+	+

1.5 Постановка задачі на проектування

Проаналізувавши та вивчивши існуючі інформаційні системи підприємств, представлених на міжнародному ринку, зрозуміло що вони досить гнучкі в налаштуванні, але не задовольняють усіх вимог Кондитерського Дому «ВАЦАК». Тому, ефективніше буде створити індивідуальну систему основу на основних потребах та вимогах для

підвищення якості функціонування відділу збуту Кондитерського Дому «ВАЦАК».

Необхідність створення інформаційної системи полягає у наступному:

- Створення БД та запис до нього даних у певному форматі.
- Читання всіх даних із БД та їх відображення.
- Додавання нового елемента даних до БД.
- Видалення будь-якого елемента даних із БД.
- Забезпечення швидкої роботи з даними;
- Виконання великих обсягів робіт по збуту готової продукції;
- Оновлення будь-якого елемента даних у БД.
- Раціональний розподіл функцій між користувачем і комп'ютером;
- Ведення обліку готової продукції;

РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ

2.1. Загальні положення

2.1.1. Найменування підсистеми: система інформаційної підтримки відділу збуту Кондитерського Дому «ВАЦАК»,

2.1.2. Результати робіт зі створення підсистеми оформлюються згідно з вимогами ДСТУ на відповідні етапи розробки. Порядок оформлення і передачі результатів у даному випадку визначається змістом і календарним планом виконання розробки.

2.2. Призначення і цілі створення підсистеми

2.2.1. Призначення підсистеми

Система призначена для інформаційної підтримки відділу збуту Кондитерського Дому «ВАЦАК» є підвищення ефективності виконання менеджерами підприємства їхніх функціональних обов'язків, а також покращення системи звітності та кількості виробленої продукції, за рахунок покращення якості виконання:

- управлінських,
- обліково-організаційних,
- звітних та інших задач.

2.2.2. Цілі створення підсистеми.

Основною метою створення підсистеми є забезпечення реалізація та наступне впровадження системи інформаційної підтримки відділу збуту Кондитерського Дому «ВАЦАК». Для досягнення поставленої мети потрібно виконати наступні задачі:

- вивчити теорію управління відділом збуту;
- вивчити проблеми організації та управління;

- проаналізувати діяльність викладача менеджерів та побудувати процесну модель;
- спроектувати інформаційну систему;
- частково реалізувати програмне забезпечення системи;
- отримати і проаналізувати результати проробленої роботи.

2.3. Характеристика об'єкта автоматизації.

2.3.1. Короткі відомості про об'єкт автоматизації.

Об'єктом автоматизації є діяльність відділу збуту Кондитерського Дму «ВАЦАК»

2.4. Вимоги до підсистеми

2.4.1. Вимоги до підсистеми в цілому.

2.4.1.1. Вимоги до структури і функціонування підсистеми.

2.4.1.1.1. підсистема повинна мати клієнтську архітектуру, що використовує базу даних (надалі — БД).

Згідно з функціональною структурою приймальної комісії, підсистема повинна бути пов'язана в мережі з автоматизованими робочими місцями:

- старшого менеджера;
- менеджера з продаж;
- менеджера;

2.4.1.1.2. Діагностування функціонування підсистеми в мережі КД «ВАЦАК» має передбачати виявлення відхилень від нормального процесу розв'язання задач і порушень у роботі комп'ютерно-технічних засобів, а також програмних помилок, забезпечуючи користувачів відповідними діагностичними повідомленнями.

Взаємозв'язок має здійснюватися на інформаційному рівні через БД із використанням технічних засобів локальних комп'ютерних мереж.

2.4.1.1.3. Розвиток і модернізація підсистеми повинні проводитися шляхом уточнення, нарощування чи заміни виконуваних функцій, модернізації технічних і програмних засобів по мірі розробки і впровадження нових поколінь комп'ютерів. Структура і технологія програмного забезпечення підсистеми повинні забезпечити простоту їх модернізації та розвитку, з можливістю збільшення розмірності задач і масивів інформації, а також можливості реалізації їх на нових ПК.

Програмно-технічні засоби функціонування підсистеми повинні мати програми з економіко-математичними та статистичними методами, методами моделювання, а також засоби табличного, текстового відображення даних. Програмна та інформаційна сумісність має забезпечуватися загальносистемним протоколом обміну, використанням проблемно-орієнтованих пакетів прикладних програм міжмашинних зв'язків і єдиною підсистемою класифікації і кодування.

2.4.1.1.4. Функціонування підсистеми має забезпечувати діалогову та мережну (розподілену) обробку даних.

2.4.1.2. Вимоги до чисельності і кваліфікації персоналу.

2.4.1.2.1. Персонал, який використовує автоматизовану підсистему, повинен дотримуватися наступних вимог:

- пройти навчання і отримати навички роботи на ПК;
- дотримуватись технологічних інструкцій при роботі з підсистемою в діалоговому режимі;
- дотримуватись умов експлуатації ПК у відповідності з інструкціями по експлуатації;
- дотримуватись правил зберігання інформації і організації резервних копій БД;
- дотримуватись правил техніки безпеки при роботі на ПК.

2.4.1.2.2. Користувачами підсистеми може виступати старший менеджер, менеджер з продаж, менеджер. Вхід у підсистему повинен здійснюватися через пароль.

2.4.1.3. Показники призначення.

2.4.1.3.1. Відповідно до п. 2.1, показники призначення повинні характеризувати ступінь та якість автоматизації планової, інформаційно-облікової і управлінської діяльності відділу збуту для його оптимального функціонування. Перелік і допустимі значення показників, при яких зберігається цільове призначення підсистеми, повинні бути визначені на стадії техноробочого проектування.

2.4.1.3.2. Підсистема повинна мати можливість налаштування на параметри об'єкта управління та периферійного обладнання при її модернізації та розвитку, а також зміні процесів та методів організаційного управління.

2.4.1.4. Вимоги до надійності.

2.4.1.4.1. Підсистема є багатофункціональною і призначена для використання протягом робочого дня. Всі функції підсистеми виконуються дискретно. У відповідності з ДСТУ 2226-93 оцінка надійності проводиться по кожній функції окремо. Враховуючи особливості функціонування підсистеми, показники її надійності є показниками надійності СУБД, на якій вона реалізована, та технічних засобів, на яких вона експлуатується. Основними показниками надійності є:

L_i — ймовірність безвідмовного виконання задачі в заданий термін (імовірність того, що i -тий запит буде виконаний);

K_r — коефіцієнт готовності ПТК (програмно-технічного комплексу);

T_v — середній час відновлення ПТК;

T_e — мінімальний час між двома відмовами за календарний місяць.

4.1.4.2. Комплекс технічних засобів повинен передбачати:

- можливість запуску і розв'язання функціональних задач із різних робочих станцій;
- можливість переходу на локальний режим роботи.

Для забезпечення надійності програмного та інформаційного забезпечення необхідно передбачити використання:

- модульного, структурного і об'єктно-орієнтованого програмування;
- програмних засобів контролю вхідної інформації з видачею користувачу повідомлень про виявлені помилки;
- програмних засобів коригування для виявлення і виправлення помилок у БД;
- засобів захисту від збоїв, несанкціонованого доступу, помилкових дій персоналу і т.д.;
- резервних копій БД.

2.4.1.5. Вимоги до безпеки.

Для забезпечення безпеки при експлуатації, налагодженні, монтажі, обслуговуванні і ремонті технічних засобів підсистеми потрібно дотримуватись вимог ДСТУ: ДСТУ 2293-99, ДСТУ ISO 6309:2007, ДСТУ 12.0.230:2008, ДСТУ 7237:2011, ДСТУ 7238:2011, ДСТУ 7239:2011; по доступним рівням освітленості, вібраційних і шумових навантажень слід дотримуватися вимог відповідно ДСТУ Б А.3.2-15:2011, ДСТУ EN 14253:2018, ДСТУ 2867-94.

2.4.1.6. Вимоги з ергономіки та технічної естетики.

Загальні ергономічні і естетичні вимоги до підсистеми повинні відповідати держстандартам ДСТУ 8604:2015, ДСТУ 7298:2013. Освітленість робочого місця повинна відповідати ДСТУ EN 12464-1:2016, ДБН В.2.5-28-2006.

Засоби відображення повинні розміщуватися таким чином, щоб кут спостереження екрану складав не більше, ніж 45 градусів, мінімальна відстань спостереження екрану — 0,3 м, рекомендована — 0,5 м.

При розробленні ПЗ слід створити зручний інтерфейс для запобігання втомлюваності користувача.

2.4.1.7. Вимоги по експлуатації, технічного обслуговування, ремонту і зберігання компонентів підсистеми.

2.4.1.7.1. Види обслуговування підсистеми визначаються у відповідності з ДСТУ EN 13306:2019. Загальні вимоги по експлуатації, технічному обслуговуванню і ремонту повинні відповідати ДСТУ 3576-97.

2.4.1.7.2. Для розміщення технічних засобів підсистеми необхідні площі, визначені в ДБН В.2.2-9-2009. При цьому слід дотримуватися вимог, зазначених в експлуатаційній документації. Напруга живлення технічних засобів підсистеми 220/380 В змінного струму, частотою (50±1) Гц. Допустиме відхилення напруги від +10 до -15%, тривалість перерв у живленні не повинна перевищувати 0,001 с.

2.4.1.7.3. Кількість, кваліфікація і режими роботи обслуговуючого персоналу повинні відповідати рекомендаціям, зазначеним в технічних умовах і інструкціях з експлуатації окремих ТЗ.

2.4.1.7.4. Склад, розміщення і умови зберігання компонентів технічних засобів підсистеми визначається рекомендаціями, зазначеними в експлуатаційній документації на ці елементи.

2.4.1.7.5. Регламент обслуговування повинен відповідати їх рівню і умовам роботи, щоб у випадку відмови підсистеми забезпечити роботу в аварійному режимі.

2.4.1.8. Вимоги до захисту інформації від несанкціонованого доступу.

Для надійності збереження і доступу до інформації необхідно використовувати засоби захисту:

- 1) серверних операційних підсистем Windows
- 2) локальної мережі та програми захисту в мережі Firewall.
- 3) клієнт-серверної СУБД:
 - тригери, представлення;

- процедури та функції;
- встановлення груп користувачів і ролей використання.

Крім цього, кожен сеанс роботи підсистеми має розпочинатися з введення індивідуального паролю. підсистема паролного захисту повинна мати власні засоби періодичної зміни паролів або використовувати стандартні засоби середовища розроблення. Для надійного захисту від несанкціонованого доступу кожен із працівників повинен мати персональний пароль. Крім того, деякі таблиці треба захистити від можливого редагування, доповнення чи вилучення інформації.

2.4.1.9. Вимоги щодо збереження інформації при аваріях.

2.4.1.9.1. Необхідно передбачити засоби резервного збереження БД в архіві після коригування і можливість завантажити БД з архіву у випадку її руйнування.

2.4.1.9.2. Резервний архів і БД мають знаходитись на різних машинних носіях чи пристроях.

2.4.1.10. Вимоги по захисту від впливу зовнішніх діянь.

2.4.1.10.1. Електрична складова електромагнітного поля завад в приміщеннях не повинна перевищувати $0,3 \text{ В/м}^2$ в діапазоні частот від 0,15 до 300 МГц. Для захисту від впливу електромагнітних полів та індустріальних завад слід передбачити різноманітні екрани та фільтри.

2.4.1.10.2. Засоби, які виключають вплив шкідливих факторів на функціонування комплексу технічних засобів, повинні бути запроєктовані згідно з ДБН В.2.2-9-2009. Обчислювальні засоби по стійкості до зовнішніх впливів повинні відповідати ДСТУ 2506-94.

2.4.1.11. Вимоги до патентної чистоти.

При створенні даної підсистеми патентні дослідження не проводяться.

2.4.1.12. Вимоги по стандартизації і уніфікації.

У підсистемі кодування інформації необхідно проводити за світовим класифікатором і стандартом.

2.4.2. Вимоги до функцій.

2.4.2.1. Перелік функцій із зазначенням вхідної та вихідної інформації наведено в таблиці 1.

Функції мають забезпечити раціональну організацію роботи користувача на основі безперервної технології: заповнення БД, довідників, формування різномірних звітів і виконання інших функцій, визначених чинним документом. При цьому пріоритетом є зручність введення та використання інформації користувачем за рахунок формування підказок і меню на екрані монітора.

Таблиця 1. Перелік функцій, вхідної та вихідної інформації

№ п/п	Найменування функції	Вхідна інформація	Вихідна інформація
1	Формування та виведення каталогу менеджерів	Таблиці БД «Менеджери», «Торвати»	Форма зі списком товарів та керівників
2	Формування та виведення каталогу товарів	Таблиці БД «Менеджери», «Товари», «Активні договори»	Форма зі списком товарів
3	Пошук серед договорів	Запит користувача, «Активні договори»	Виділення потрібного договору

2.5. Вимоги до видів забезпечення.

2.5.1. У вимогах до математичного забезпечення (МЗ) підсистема не вимагає спеціального математичного забезпечення для реалізації покладених на неї функцій. Достатньо можливостей обраної СУБД.

2.5.2. Вимоги до інформаційного забезпечення (ІЗ).

2.5.2.1. Інформаційне забезпечення підсистеми повинно містити дані, достатні для виконання всіх покладених на підсистему функцій. ІЗ повинно гарантувати раціональну організацію зберігання інформації та доступу до неї.

Заповнення БД інформацією покладається на замовника за методиками і формами, створеними розробниками підсистеми.

Склад, структура і спосіб організації інформації представляються у логічній моделі БД і можуть уточнюватись на етапі технічного проектування.

2.5.2.2. Слід передбачити захист даних від руйнування при аваріях і порушеннях у енергоживленні підсистеми — використання резервних копій БД.

2.5.3. Вимоги до лінгвістичного забезпечення (ЛЗ).

2.5.3.1. Для розробки програмних засобів, які реалізують виконання функцій і забезпечують сервіс користувачів повинні використовуватися мови високого рівня, які забезпечують створення структурних програм, а також мова обраної СУБД для здійснення доступу та маніпулювання даними.

2.5.3.2. Організація діалогу користувача до підсистеми має будуватися на наборах меню і підказок, орієнтованих на виконання користувачем функцій. Запити користувача до підсистеми повинні задаватись переважно природною мовою.

2.5.4. Вимоги до програмного забезпечення (ПЗ).

2.5.4.1. Загальносистемне ПЗ має забезпечувати надійне і якісне виконання функціональних завдань підсистеми. До загальносистемного ПЗ належить:

- операційна підсистема (далі ОС) — Windows ;
- система управління БД (далі СУБД) — MySQL 7.

2.5.4.2. Загальні вимоги до підсистемного ПЗ можна сформулювати так:

- мінімальні вимоги до ресурсів технічних засобів (ТЗ);
- максимальна швидкодія;
- повне задоволення потреб функціональних завдань підсистеми.

2.5.4.3. Вимоги до ОС:

- мінімальне використання ресурсів комп'ютера для власних потреб, передусім оперативної і дискової пам'яті;
- максимальна швидкодія при управлінні зовнішніми пристроями;
- ОС клієнта — Windows 7

2.5.4.4. Вимоги до СУБД:

- максимальне задоволення потреб функціональних задач;
- надійність;
- ефективне управління потрібного обсягу і структури;
- швидкість виконання запитів користувачів;
- мінімальні вимоги до ТЗ.

2.5.4.5. Програмні засоби введення та виведення даних і ведення діалогу повинні забезпечувати:

- виведення необхідних даних на екран у вигляді відповідних відеограм;
- супровід введення даних контролем і сигналізацією користувачу про наявність помилок з можливістю їх виправлення під час введення даних;
- керований комп'ютером діалог при введенні даних;
- виведення даних у відповідному вигляді (формі документа) за запитом користувача.

2.5.4.6. При розробленні спеціального ПЗ слід виконати наступні вимоги:

- використані програми мають бути сумісні між собою та із загальносистемним ПЗ;
- ПЗ має розроблятися засобами об'єктно-орієнтованого програмування;
- забезпечити відповідність інтерфейсу користувача стандартам Windows;

- необхідна модульна структура програм;
- повинна бути передбачена можливість розширення складу задач у відповідності з новими функціональними потребами;
- ПЗ не повинно залежати від типу зовнішніх пристроїв (принтерів, дисків, сканерів тощо);
- діалог із користувачем повинен проводитись за допомогою клавіатури або миші з поясненням виконання дій і можливістю отримання підказки.

2.5.5. Вимоги до технічного забезпечення.

2.5.5.1. Технічні засоби підсистеми (табл. 2) повинні забезпечувати виконання функцій, перерахованих в таблиці 1.

2.5.5.2. Засоби обчислювальної техніки повинні забезпечувати обмін інформації в об'ємах, приведених в п. 2.5.2.

Таблиця 2. Вимоги до технічного забезпечення підсистеми

№ п/п	Основні характеристики комп'ютера
Технічне забезпечення для сервера	
1	HP ML115 Intel Xeon Quad Core 2,5 GHz\8 Gb\1 TB RAID5\ LAN 1 Gbit
Технічне забезпечення для клієнта	
1	Athlon QL-65 Dual Core 2,1 GHz; RAM: 2048 Mb; HDD: 250 Gb;
2	Монітор 15"
3	Миша USB
4	Клавіатура USB

2.5.6. Вимоги до метрологічного забезпечення.

Система не має вимірювальних каналів, вимірювального обладнання і приладів, тому вимоги до даного виду забезпечення не висуваються.

2.5.7. Вимоги до організаційного забезпечення.

2.5.7.1. Організаційне забезпечення підсистеми розробляється в відповідності з вимогами державного стандарту по АСУП.

2.5.7.2. При впровадженні підсистеми не передбачається збільшення штатної чисельності підприємства. Територіальне розміщення робочих місць, на яких буде встановлена підсистема, визначається підприємством.

2.5.7.3. До функціонування підсистеми висуваються наступні вимоги:

- наказом директора визначається список співробітників, які мають доступ до підсистеми;
- контроль і прийняття рішень при аварійних ситуаціях при експлуатації підсистеми здійснює відповідальний за підсистему.

2.6. Склад і зміст робіт по створенню підсистеми.

2.6.1. Стадії створення підсистеми і терміни виконання робіт наведені в таблиці 3.

Таблиця 3. Найменування робіт при створенні підсистеми

№ п/п	Найменування робіт	Строки виконання робіт
1	Передпроектне дослідження об'єкта автоматизації	25.09.2020
2	Технічне завдання	25.10.2020
3	Технічний проект	10.11.2020
4	Оформлення документації	20.12.2020

2.7. Порядок контролю і приймання підсистеми

2.7.1. Підсистема вводиться на діючому КД «ВАЦАК». При введенні в дію підсистема повинна пройти приймальні випробування згідно з ДСТУ 3974-2000.

2.7.2. Випробування для визначення працездатності і рішення про можливість приймання підсистеми в дослідну експлуатацію проводять розробники разом із замовником. Програму випробувань складає розробник і затверджує замовник.

2.7.3. Здача в дослідну експлуатацію здійснюється на основі технічного завдання та інструкції користувача. За результатами дослідної експлуатації формується перелік доробок і рекомендовані строки їх виконання.

2.7.4. Введення в дію підсистеми оформлюється актом здачі-прийому.

2.8. Вимоги до складу і змісту робіт із підготовки до введення підсистеми в дію.

Для введення в дію замовник виконує ряд робіт із підготовки об'єкта:

- проводить укомплектування технічних засобів;
- організовує навчання користувачів підсистеми роботі на ПК і вивчення інструкції з її експлуатації;
- проводить дослідну експлуатацію і вводить підсистему в дію.

2.9. Вимоги до документації.

2.9.1. На підсистему розробляється комплекс документації у складі: технічне завдання та технічний проект.

2.9.2. Документація на підсистему розробляється у відповідності з вимогами Державних стандартів серії 19 «Єдина підсистема програмної документації» та серії 24 «Єдина підсистема стандартів автоматизованих підсистем управління».

2.10. Джерела розробки

2.10.1. При розробленні технічного завдання на підсистему використано наступні документи:

- ДСТУ 3973–2000 підсистема розроблення та поставлення продукції на виробництво;
- ДСТУ 3008-2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання;
- ДСТУ Б В.2.5–82:2016 Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом.

РОЗДІЛ 3. РОЗРОБКА КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ

3.1 Обґрунтування вибору програмного забезпечення використовуваного для розробки системи

3.1.1 Структура програмного забезпечення

Visual Studio – інтегроване середовище розробки програмного забезпечення від фірми Microsoft. Дане середовище дозволяє створювати різноманітні програмні продукти: консольні програми, програми з графічним інтерфейсом, наприклад віконні додатки Windows Forms, а також Web-додатки тощо[7].

Платформа .NET Framework - це технологія, яка підтримує створення і виконання веб-служб і додатків Windows. При розробці платформи .NET Framework враховувалися наступні цілі[8].

- Забезпечення узгодженої об'єктно-орієнтованого середовища програмування для локального збереження і виконання об'єктного коду, для локального виконання коду, розподіленого в Інтернеті, або для віддаленого виконання.
- Надання середовища виконання коду, в якій:
 - зведена до мінімуму ймовірність конфліктів в процесі розгортання програмного забезпечення і управління його версіями;
 - гарантується безпечне виконання коду, включаючи код, створений невідомим або не повністю довіреною стороннього постачальника;
 - виключаються проблеми з продуктивністю середовищ виконання скриптів або інтерпретується коду;
- забезпечуються єдині принципи розробки для різних типів додатків, таких як додатки Windows і веб-додатки;

- забезпечується взаємодія на основі промислових стандартів, яке гарантує інтеграцію коду платформи .NET Framework з будь-яким іншим кодом.

Платформа .NET Framework складається з загальномовного середовища виконання (середовища CLR) і бібліотеки класів .NET Framework. Основою платформи .NET Framework є середовище CLR. Середу виконання можна вважати агентом, який керує кодом під час виконання і надає основні служби, такі як управління пам'яттю, управління потоками і віддалене взаємодія. При цьому середовищем накладаються умови суворої типізації та інші види перевірки точності коду, що забезпечують безпеку і надійність. Фактично основним завданням середовища виконання є управління кодом. Код, який звертається до середовища виконання, називають керованим кодом, а код, який не обертається до середовища виконання, називають некерованим кодом. Бібліотека класів є комплексною об'єктно-орієнтованою колекцією повторно використовуваних типів, які застосовуються для розробки додатків - починаючи з звичайних додатків, що запускаються з командного рядка, і додатків з графічним інтерфейсом (GUI) і закінчуючи додатками, що використовують останні технологічні можливості ASP.NET, такі як веб-форми і веб-служби XML[8].

C # - це об'єктно-і компонентно-орієнтований мову програмування. C # надає мовні конструкції для безпосередньої підтримки такої концепції роботи. Завдяки цьому C # підходить для створення і застосування програмних компонентів. З моменту створення мову C # збагатився функціями для підтримки нових робочих навантажень і сучасними рекомендаціями по розробці ПЗ[9].

MS SQL Server - це платформа для вирішення критично важливих завдань в масштабі підприємства, що володіє високою доступністю, підвищеною продуктивністю і безпекою. Рішення являє собою добре

масштабується, повністю реляційний, швидкодіючий сервер, здатний обробляти великі обсяги даних для клієнт-серверних додатків.

Функції SQL Server, що дозволяють забезпечити дані і боротися з погрозами, розширюються і удосконалюються від версії до версії. В основу покладено принцип мінімальних привілеїв, доступ до інформації через збережені процедури і визначені користувачем функції, що забезпечують доступ тільки до дозволених ресурсів. Прозоре шифрування даних і журналів (Transparent data encryption, TDE), шифрування на рівні осередків і шифрування з'єднань захищають інформацію від крадіжки. Плюс розміщення файлів бази на Encrypting File System. Однак нові загрози і завдання для SQL-сервера виникають постійно, і інструменти безпеки не завжди з ними справляються[10].

3.1.2 Опис логічної та фізичної моделі бази даних розробки

Моделі даних можна створювати кількома способами. Можливо, вам доведеться створити модель даних із об'єктів, які існують у вашій схемі бази даних, наприклад. AllFusion ERwin DM дозволяє розробникам баз даних інтегрувати об'єкти баз даних між фазою проектування та фазою реалізації життєвого циклу розробки бази даних. Продукт підтримує більше десятка популярних СУБД.

Тестування AllFusion ERwin DM за допомогою SQL Server 2000 та Oracle 9i засвідчило, що інтеграція була безперебійною. Після того, як ми запустили просту процедуру налаштування, AllFusion ERwin DM встановив зв'язок із сервером бази даних і безперешкодно переміщував об'єкти бази даних між двома середовищами.

erwin Data Modeler (erwin DM) - це нагородами інструмент моделювання даних, який використовується для пошуку, візуалізації, проектування,

розгортання та стандартизації високоякісних корпоративних даних. Відкривайте та документуйте будь-які дані з будь-якого місця для забезпечення послідовності, чіткості та повторного використання артефактів за допомогою широкомасштабної інтеграції даних, управління основними даними, управління метаданими, великих даних, бізнес-аналітики та аналітичних ініціатив - і все це при підтримці управління даними та розвідувальних зусиль[12].

Ви можете так само легко генерувати SQL об'єктів бази даних і будувати базу даних лише за кілька кліків миші. Ви також можете зберегти згенерований SQL як файл .sql та змінити його відповідно до потреб вашого магазину. Сформований SQL не гарний, але функціональний. Створений нами зразок SQL працював бездоганно, навіть з рекурсивними відносинами, обмеженнями перевірки та складеними первинними ключами. CA заявляє, що його продукт також підтримує всі синтаксис Oracle 10g та об'єкти баз даних.

Логічний, чіткий макет AllFusion ERwin DM дозволяє легко створювати модель бази даних з нуля. Діаграми взаємозв'язків сутності (моделі даних, що представляють логічний вигляд бази даних та ідентифікують взаємозв'язки між сутностями), можна створювати без посилання на велику документацію продукту. Просто відкрийте новий документ, визначте тип моделі - логічну, фізичну або обидві - перетягніть піктограму сутності з панелі інструментів і помістіть її в робочу область. Клацніть правою кнопкою миші на сутності, щоб побачити список варіантів. Виберіть параметр атрибутів, і з'явиться акуратне діалогове вікно з опціями додавання, зміни та видалення атрибутів, а також встановлення первинного ключа. Програмне забезпечення також постачається з шаблонами, які ви можете використовувати для створення своєї моделі, або ви можете отримати нову модель даних із існуючої.

Зв'язок з BPWin (моделювання даних в стандарті IDEF0, IDEF3, DFD, SADT) дозволяє скоротити час проектування і розробки складних інформаційних систем[13].

3.2 Реалізація логічної і фізичної бази даних

Логічна модель бази даних складається з таких сутностей:

- Менеджери;
- Продукція
- Інформація про активні договори;

Детальний опис сутностей наведений у табл.3.1.

Таблиця 4. «Структура бази даних»

Назва сутності	Список полів	Тип поля	Опис сутності
Продукція	Код продукції	Varchar(70)	
	Назва	Varchar(70)	
	Кількість	integer	
Менеджери	Посада	Varchar(70)	
	П.І.Б.	Varchar(70)	
	Код співробітника	integer	
	Товар	Varchar(70)	
	Підпорядковується	Varchar(70)	
Активні договори	Товар	Varchar(70)	
	№ договору	integer	
	№ заявки	integer	
	Назва	Varchar(70)	
	Площа яку займає груз	integer	
	Ширина грузу	integer	
	Висота грузу	integer	
	Довжина грузу	integer	

	Ящиків	integer	
	Кількість	integer	
	Коментар до договору	Varchar(70)	

Застосовано ідентифікований («один-до-багатьох» - одному екземпляру батьківської сутності відповідає декілька екземплярів дочірньої сутності) та неідентифікований зв'язок («один-до-одного», «один-до-багатьох»). Зв'язок відбувається за допомогою первинних ключів.

База даних повинна відповідати вимогам нормалізації даних, які мають забезпечити компактність зберігання даних. Основна ідея полягає в тому – кожен факт повинен зберігатися лише в одному місці, що дозволяє уникнути надлишку інформації в базі даних що будується.

Фізичну модель бази даних згенеровано у MS SQL 2016. Схема бази даних представлена у Додатку В

3.3. Інформаційне забезпечення розробки Microsoft SQL 2016

Microsoft SQL Server - це платформа для вирішення критично важливих завдань в масштабі підприємства, що володіє високою доступністю, підвищеною продуктивністю і безпекою[14].

Функції SQL Server, що дозволяють убезпечити дані і боротися з погрозами, розширюються і удосконалюються від версії до версії. В основу покладено принцип мінімальних привілеїв, доступ до інформації через збережені процедури і визначені користувачем функції, що забезпечують доступ тільки до дозволених ресурсів. Прозоре шифрування даних і журналів (Transparent data encryption, TDE), шифрування на рівні осередків і шифрування з'єднань захищають інформацію від крадіжки. Плюс розміщення

файлів бази на Encrypting File System. Однак нові загрози і завдання для SQL-сервера виникають постійно, і інструменти безпеки не завжди з ними справляються.

У SQL Server 2016 з'явилося три великих нововведення, що дозволяють захистити дані від дій недобросовісних співробітників, гнучко розмежувати доступ і приховати дані від адміністратора сервера. Це Always Encrypted, Row-Level Security і Dynamic Data Masking.

TDE шифрує дані всієї бази в стані спокою, але системний адміністратор без проблем може отримати доступ до даних, що зберігаються в базі. Функція Always Encrypted, доступна в Enterprise (і Developer), дозволяє шифрувати і дешифрувати дані всередині клієнтських додатків, не розкриваючи ключів шифрування SQL Server. Дані шифруються на рівні стовпця як в спокої, так і в пам'яті. Тобто з Always Encrypted забезпечується чіткий поділ між власниками даних і тими, хто управляє даними (але не повинен мати доступу). Тепер клієнти можуть впевнено зберігати конфіденційну інформацію на сторонніх серверах, не турбуючись про витік.

Розшифрування проводиться з використанням розширеної клієнтської бібліотеки. Це призводить до важливого обмеження Always Encrypted - бібліотеки повинні підтримувати клієнт. Фактично сьогодні з Always Encrypted працює єдина бібліотека - ADO.NET 4.6 (.NET Framework 4.6). Також підтримуються не всі види даних і деякі функції (пошук, реплікація, розподілені запити, тригери частково). Повний список доступний по посиланню. Використання Always Encrypted позначиться також на продуктивності і зажадає більше місця під зберігання зашифрованих даних.

3.4 Інструкція користувача системи

3.4.1 Початок роботи і головне меню системи

Для початку роботи з системою, достатньо просто зробити подвійний клік по іконці програми, після чого відкриється головне меню системи де потрібно буде увести логін та пароль, Рис. 3.1

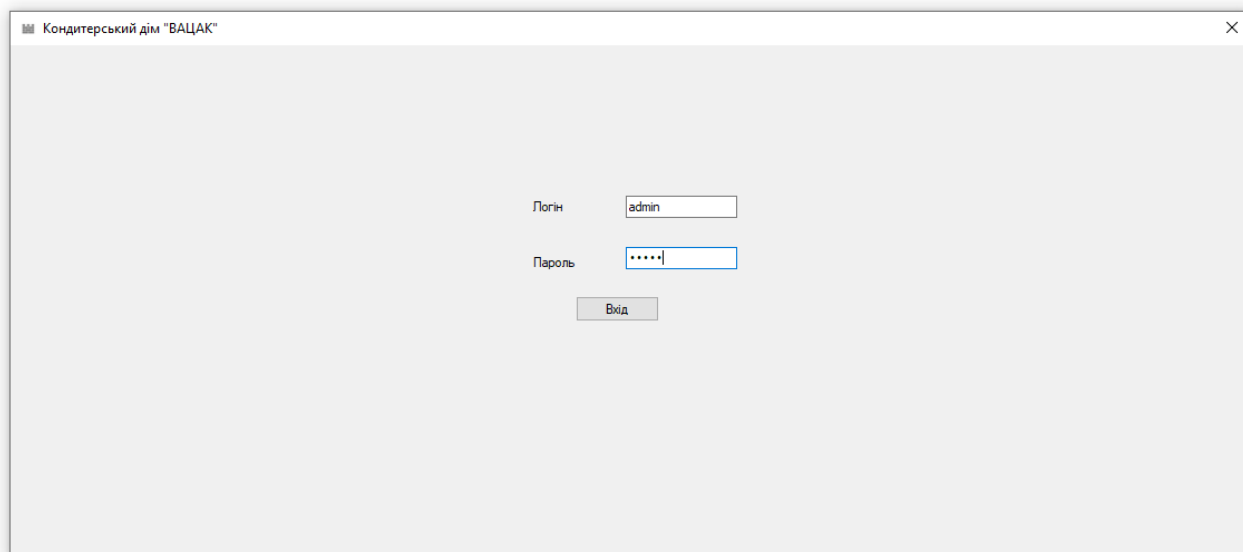


Рис. 3.1 Головне вікно системи

3.4.2 Форми введення і перегляду

Після введення логіну та паролю відкриється головне вікно програми, а саме вкладка на якій показано менеджерів які працюють по збуту товару . На Рис. 3.2 наведено приклад відкритого пункту «Менеджери»

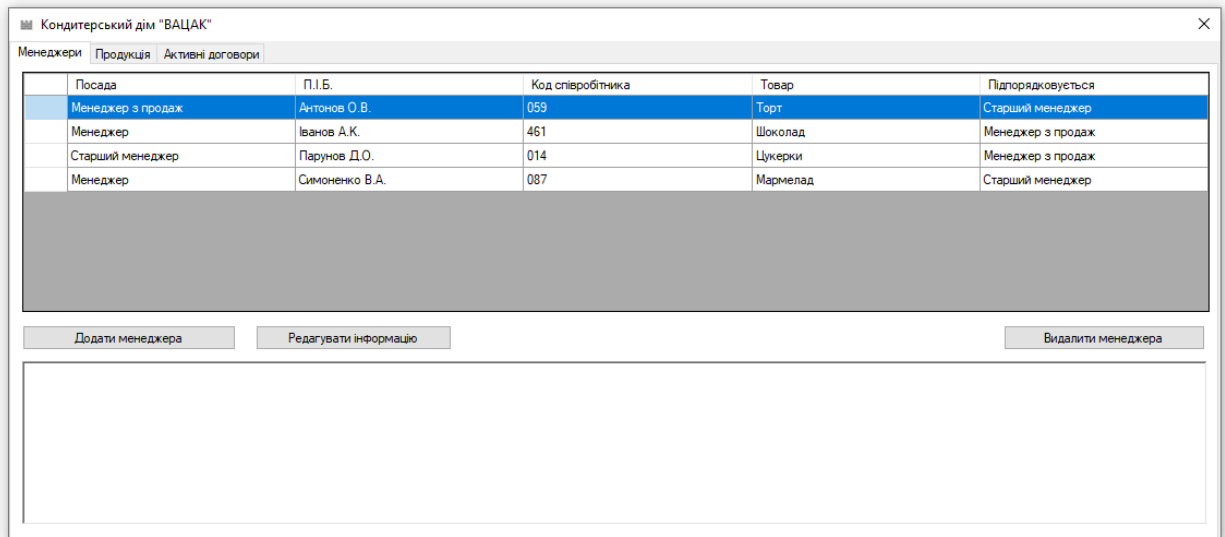


Рис. 3.2 Відкрите вікно «Менеджери»

На Рис. 3.3 наведено приклад відкритого пункту «Продукція»

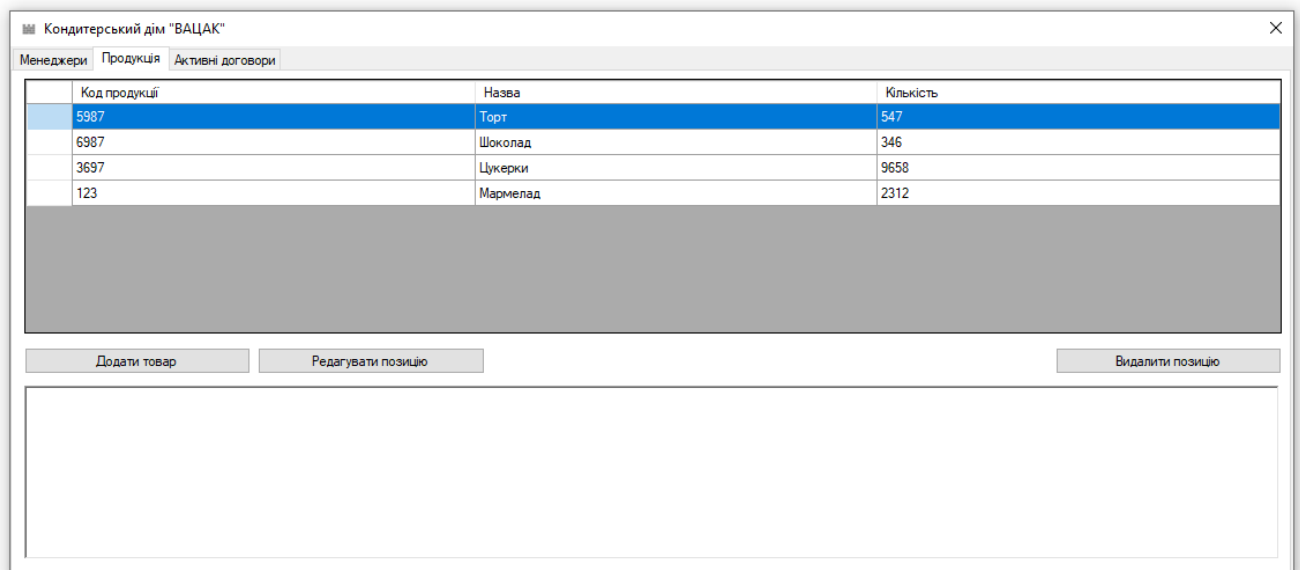


Рис. 3.3 Відкрите вікно «Продукція»

На Рис. 3.4 наведено приклад відкритого пункту «Активні договори», з якого можна перейти до вікна «Активні договори» (Рис. 3.5)

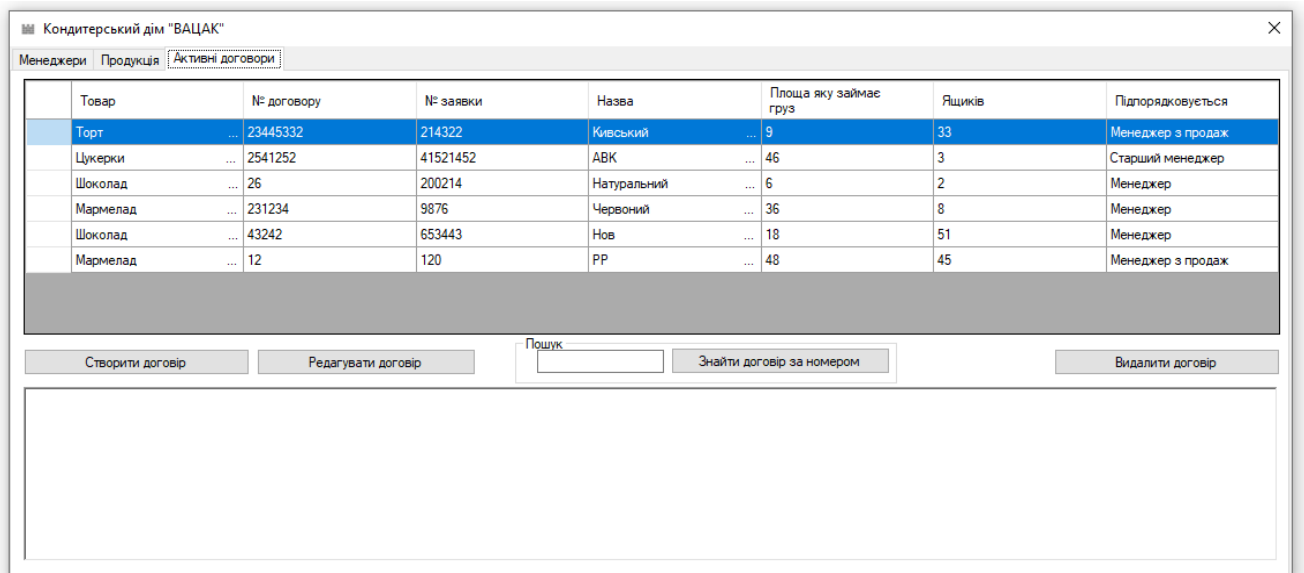


Рис. 3.4. Вікно "Активні договори"

3.4.3. Виконання пошуку

Виконати пошук договору можна у вкладці «Активні договори», приклад наведено на Рис. 3.5

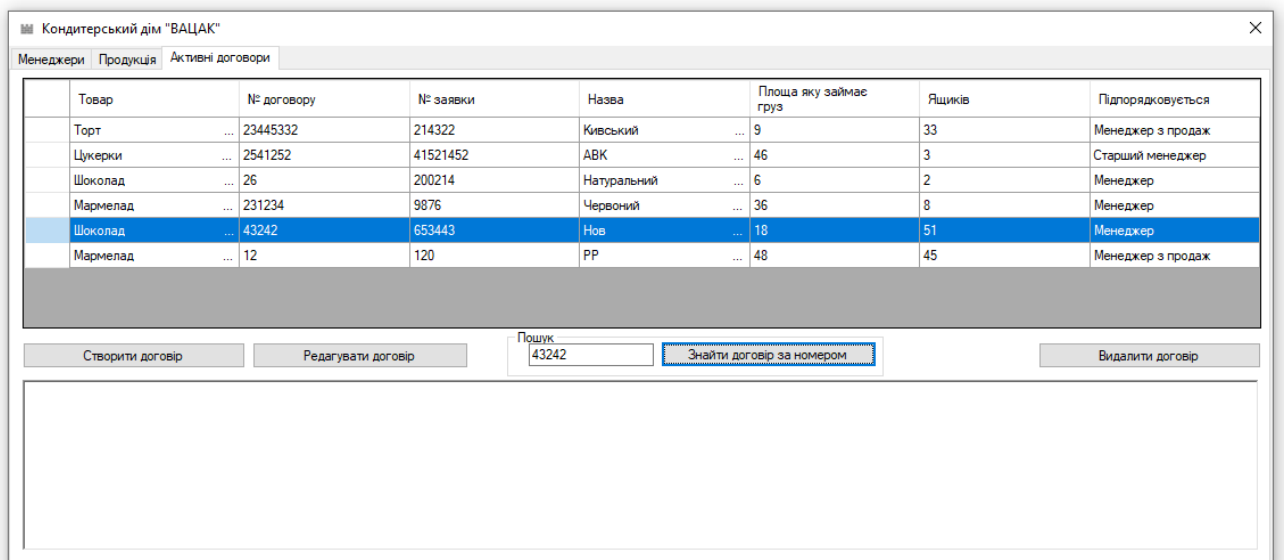


Рис. 3.5. Результат виконання пошуку

3.5 Економічна ефективність використання підсистеми

Техніко-економічний ефект від впровадження інформаційної системи визначається за співвідношенням витрат та розробку системи, її встановлення у відповідному експлуатації.

1. Ступінь новизни розроблюваних задач – "Г" – використання типових проектних рішень за умови їх змін.
2. Група складності алгоритму – 3.
3. Узагальнені дані вхідної та вихідної інформації для інформаційної системи підтримки діяльності відділу збуту за видами вхідної та вихідної інформації табл. 2.2

За таблицею 2.2 визначається вид інформації.

Таблиця. 3.5

Вид інформації	Позначення	К-сть наборів даних
Змінна інформація	ЗІ	m=2
Нормативно-довідкова інформація	НДІ	n=2
Банк (база) даних	БД	p=1
Обробка в режимі реального часу	РЧ	Так
Забезпечення телекомунікаційної обробки даних і управління віддаленими об'єктами	ТОУ	Ні

Витрати часу на розробку інформаційної систему для відділу збуту ТОВ «ВАЦАК» табл. 2.3, а саме на розробку ескізного проекту і передпроектного дослідження Т1 і технічного завдання Т2, будуть наступні.

Таблиця 3.6.

Вид системи	Стадія розробки системи	
	Передпроектне дослідження	Технічне завдання
	Г	Г
Управління матеріально-технічним постачанням, управління збутом продукції, управління комплектацією.	$t_1 = 35$	$t_2 = 30$

Визначається базове значення витрат часу для стадій "Технічний проект" , "РП" і "Впровадження". Для цього використовуються дані для системи якості.

Вхідними даними для визначення є:

- кількість форм вхідної інформації $B_1 = 2$,
- кількість форм вихідної інформації $B_2 = 3$,
- базове значення витрат часу для стадій "Технічний проект": $T_{B3} = 59$;
- базове значення витрат часу для стадій "РП": $T_{B4} = 103$;
- базове значення витрат часу для стадій "Впровадження": $T_{B5} = 40$;

Базове значення витрат часу T_B коригується за допомогою поправочних коефіцієнтів для всіх стадій розробки автоматизованої системи.

- Розрахунок витрат часу для стадії "Технічний проект" (T_3)

Таблиця 2.4. Коефіцієнти k_1 , k_2 , k_3 для стадії "Технічний проект".

Таблиця 3.7

Вид використаної інформації	Ступінь новизни
	B
k_1 (ЗІ)	0.5
k_2 (НДІ)	0.43
k_3 (БД)	1.25

$$k_{II} = \frac{k_1 * m + k_2 * n + k_3 * p}{m + n + p} = \frac{0.5 * 2 + 0.43 * 2 + 1.25 * 1}{2 + 2 + 1} = 0.62$$

Коефіцієнт ступеню новизни проекту, k_O , що враховує вид обробки інформації для трьох стадій розробки системи, для системи інформаційної підтримки діяльності відділу збуту значення в табл. 2.5, з урахуванням того, що обробка здійснюється в режимі реального часу табл. 2.1.

Таблиця 3.8

Стадія розробки проекту	Вид обробки	Ступінь новизни
		Г
Технічний проект	РЧ	1.1
РП	РЧ	1.15
Впровадження	РЧ	1.05

Витрати часу для стадії “технічний проект” T_3 визначається за формулою 2.1.2

$$T_3 = T_{B3} * k_{II} * k_O = 59 * 0.62 * 1.1 * 0.85 = 34.2$$

- Розрахунок витрат часу для стадії "РП" (T_4) системи інформаційної підтримки діяльності відділу збуту.

Для визначення витрат часу на стадії "РП" використовують

$$k_{II} = \frac{k_1 * m + k_2 * n + k_3 * p}{m + n + p}$$

де k_{II} – коефіцієнт, що враховує вид використаної інформації і визначається:

$$k_{\Pi} = \frac{1,1 * 2 + 0,58 * 2 + 0,48 * 1}{2 + 2 + 1} = 0,77$$

Таблиця 3.9 Коефіцієнти k1, k2, k3 для стадії "РП".

Вид використаної інформації	Група складності алгоритму	Ступінь новизни
		Г
ЗІ	3	0,48
НДІ	2	0,29
БД	1	0,24

Коефіцієнт складності контролю вхідної та вихідної інформації на стадії "РП".

$$k_c = 1,16$$

Витрати часу T_4 вимірюються в людино-днях.

$$T_4 = T_{B4} * k_{\Pi} * k_o * k_c, = 103 * 1,08 * 1,15 * 1,16 = 148,39$$

Для стадії визначення загальних витрат часу на "Впровадження" T_5 (люд-днів).

$$T_5 = T_{B5} * k_{\Pi} * k_o * k_c, = 40 * 1,08 * 1,05 * 1,16 = 53,3$$

Таким чином, загальні затрати людської праці на проектування інформаційної системи складають:

$$T_{\Sigma} = 30 + 35 + 34,2 + 148,39 + 52,6 = 300,19$$

Для дипломної роботи кількість робочих годин складає 320 із 8-годинним робочим днем, тому на розробку проекту виділено Φ , днів:

$$\Phi = \frac{320}{8} = 40 \text{ людино-днів.}$$

Для дипломної роботи $\Phi = 40$ днів. Тоді визначаємо кількість місяців із розрахунку 20 робочих днів.

Кількість місяців на розробку, M :

$$M = \frac{\Phi}{20} = \frac{40}{20} = 2$$

Отже, для виконання такого проекту потрібно таку чисельність виконавців $Ч$.

$$Ч = \frac{T_{\Sigma}}{\Phi} = \frac{300}{40} \approx 7.5$$

Якщо прийняти, що оплата програміста здійснюється в розмірі 15000 грн, то оплата праці всіх виконавців, яка підраховується складе:

$$V'_1 = Ч * M * ЗП_{\text{прог}} = 7,5 * 2 * 15000 = 225000 \text{ грн.}$$

Витрати, пов'язані з розробкою програми на ПК

1. Розрахунок річного фонду часу роботи ПК.

Дійсний річний фонд часу ПК у годинах дорівнює числу робочих годин у році для оператора, за винятком часу на технічне обслуговування і ремонт ПК (в середньому 5год/міс + 6 діб/рік).

$$T_{ПК} = 3000 - (6 \cdot 8 + 5 \cdot 12) = 2892 \text{ год}$$

Оскільки під час виконання дипломної роботи студент в середньому витрачає 400 год. машинного часу, то величина фонду часу ПК дорівнює (таблиця 2.1)

$$T'_{ПК} = 2892 \cdot \frac{400}{3000} = 376 \text{ год.}$$

Поточні витрати на експлуатацію.

Балансова вартість ПК вираховується із C_p – ринкова вартість ПК, орієнтовно 10000 грн., $k_{уН}$ – коефіцієнт, що враховує витрати на установку і налагодження ПК і дорівнює 0.12.

$$C_{ПК} = C_p \cdot (1 + k_{уН}) = 10000 \cdot (1 + 0.12) = 11200$$

Амортизаційні відрахування використання ПК, ЗАМ, обчислюється, норма амортизаційних відрахувань, яка для ПК дорівнює $HA = 5$:

$$Z_{ам} = \frac{11200}{5} = 2240$$

Витрати на електроенергію, споживану ПК, визначаються за формулою (7.14): де $P_{ПК}$ – потужність ПК, орієнтовно дорівнює 0.4 квт,

$T_{ПК}$ – фонд корисного часу роботи ПК, дорівнює 361.5 год.,

$C_{ЕЛ}$ – вартість 1квт електроенергії, дорівнює 1.68 грн/кВт,

A – коефіцієнт інтенсивного використання ПК, дорівнює 0.9.

Z_p – витрати на поточний ремонт і технічне обслуговування ПК дорівнюють 6% від $C_{ПК}$.

$Z_{МАТ}$ – непрямі витрати, пов'язані з експлуатацією ПК, дорівнюють 5% від $C_{ПК}$.

$$Z_{ЕЛ} = P_{ПК} * T_{ПК} * C_{ЕЛ} * A = 0.4 * 361.5 * 1.68 * 0.9 = 218,63$$

2.2 Поточні витрати на експлуатацію визначаються по формулі:

$$V_1'' = Z_{ЗП} + Z_{АМ} + Z_{ЕЛ} + Z_p + Z_{МАТ},$$

де $Z_{ЗП}$ – заробітна плата обслуговуючого персоналу (якщо роботи виконуються не на власному ПК). Ця величина дорівнює 7500 грн.;

$Z_{АМ}$ – амортизаційні відрахування, що обчислюються за формулою:

$$Z_{АМ} = C_{ПК} / N_A, \text{ де } C_{ПК} \text{ – балансова вартість ПК,}$$

N_A – норма амортизаційних відрахувань, яка для ПК дорівнює 5.

Таким чином маємо:

$C_{ПК} = 11200$ грн, $Z_{АМ} = 2240$ грн, $Z_{ЕЛ} = 97.6$ грн, $Z_p = 672$ грн, $Z_{МАТ} = 560$ грн.

$$V_1'' = Z_{ЗП} + Z_{АМ} + Z_{ЕЛ} + Z_p + Z_{МАТ} = 11200 + 2240 + 218,63 + 672 + 560 = 14890.63$$

Отже, загальні витрати на розробку програмного забезпечення комп'ютерної системи розраховуються за формулою.

$$V_1 = V_1' + V_1'' = 225000 + 14890.63 = 239890.63 \text{ грн.}$$

Витрати на придбання і установку ПК:

$$V_2 = 0 \text{ грн.}$$

Тому що немає потреби в купівлі ПК, то ці витрати дорівнюють "0".

Ці на підготовку приміщення залежать від стану приміщення, де буде встановлюватися ПК. Таке приміщення уже є і ремонту не потребує, тому

$$V_3 = 0.$$

В середньому навчання персоналу триватиме 1 місяць, тому можна вважати, що $V_4 = 4 \text{ чоловіка у відділі} * 4500 \text{ грн} = 18000 \text{ грн}$

Загальна вартість розробки і впровадження системи V_{Σ} , вираховується за формулою

$$V_{\Sigma} = V_1 + V_2 + V_3 + V_4 = 239890.63 + 0 + 0 + 18000 = 257890.63$$

грн.

Оскільки норма амортизаційних втрат для комп'ютерних систем $H_A = 5$, то для обрахування річного економічного ефекту слід брати до розгляду величину формули

$$V_P = \frac{V_{\Sigma}}{H_A} = \frac{257890.63}{5} = 51578.13 \text{ грн.}$$

Річний прибуток Pr від впровадження системи буде досягнуто за рахунок підвищення кількості укладання успішних договорів і орієнтовно складатиме 150000 грн на рік.

$$K_{\text{ЕФ}} = \frac{Pr}{V_P} = \frac{150000}{51578.13} = 2.9$$

Термін окупності розробки дорівнює визначається за формулою

$$T_{\text{ОК}} = \frac{1}{K_{\text{ЕФ}}} = \frac{1}{2.9} = 0.34$$

Таким чином, термін окупності інформаційної системи буде близько 4 місяців.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

Закон України “Про охорону праці” прийнятий у 1992 році. Даний закон регулює трудові відносини всіх працівників, сприяючи зростанню продуктивності праці, поліпшенню якості роботи, встановлює високий рівень умов праці. Закон передбачає створення безпечних і нешкідливих умов праці на всіх підприємствах, в установах, організаціях; обов’язки власника та працівника щодо виконання нормативних актів про охорону праці; відшкодування власником або уповноваженим ним органом шкоди працівникам у разі ушкодження їх здоров’я; контроль за додержанням вимог нормативних актів про охорону праці.

У приміщенні відсутні інфрачервоні, ультрафіолетові та електромагнітні випромінювання, адже усі монітори ПК вироблені на основі рідкокристалічної матриці, підсвітка якої здійснюється неоновною лампою, що не має сильного електромагнітного випромінювання, і є сертифікованими в Україні.

Нині існує дуже багато методів забезпечення безпеки технологічного обладнання, а з часом вони постійно будуть розширюватися й вдосконалюватися.

Методи забезпечення безпеки обладнання поділяються на:

- $7030; L=V$;
- $G0AB:>2V$.

До загальних належить механізація і автоматизація технологічних процесів, дистанційне управління і спостереження, блокування і сигналізація, надійність і міцність конструктивного виконання.

До часткових методів належать захист обладнання від певної безпеки. Це може бути герметизація, екранування, теплоізоляція, звукоізоляція, амортизація, огороження, заземлення і т. ін.

Безпека технологічного обладнання забезпечується правильним вибором методів захисту. Крім цього безпека праці забезпечується:

- 28:>@8AB0==O< C :>=AB@C:FVOE A?5FV0;L=8E 70E8A=8E 70A>1V2;
- 4>B@8<0==O< 5@3>=><VG=8E 28<>3;
- 2:;NG5==O< 28<>3 157?5:8 C B5E=VG=C 4>:C<5=B0FVN 7 <>=B06C, 5:A?;C0B0FVW, @5<>=BC V V=.

Конструктивні елементи технологічного обладнання не повинні мати гострих країв, кутів, нерівних, гарячих чи переохолоджених поверхонь.

Рухомі частини технологічного обладнання, а також пасові та ланцюгові передачі мають бути огороженні або захищені іншим шляхом якщо огороження не допускається.

Для забезпечення електробезпеки використовуються окремо або в поєднанні один з одним такі технічні способи та засоби:

- 70E8A=5 7075<;5==O;
- 70=C;5==O;
- 28@V2=N20==O ?>B5=FV0;V2;
- <0;0 =0?@C30;
- 70E8A=5 2V4V<:=5==O;
- V7>;OFVO AB@C<>?@>2>4V2;
- >3>@>46C20;L=V ?@8AB@>W;
- ?>?5@546C20;L=0 A83=0;V70FVO, 1;>:C20==O, 7=0:8 157?5:8;
- 70A>18 70E8ABC B0 70?>1V6=V ?@8AB@>W

Заземлення і занулення забезпечують спрацювання приладів захисту, швидко автоматичне вимикання пошкодженої установки від мережі.

Захисне заземлення і занулення виконують з метою:

- 70157?5G5==O =>@<0;L=8E @568<V2 @>1>B8 CAB0=>2:8;

- 70157?5G5==O 157?5:8 ;N459 ?@8 ?>@CH5==V V7>;OFVW <5@56V AB@C<>?@>2V4=8E G0AB8=;
- 70E8ABC 5;5:B@>CAB0B:C20==O 2V4 ?5@5=0?@C38;
- 70E8ABC ;N459 2V4 AB0B8G=>W 5;5:B@8:8.

Малу напругу (не більше 42 В) застосовують для живлення електроприймачів невеликої потужності: ручного електрифікованого інструменту, переносних ламп, ламп місцевого освітлення, сигналізації»

У приміщеннях без підвищеної і особливої небезпеки ' використовують переносні світильники з напругою 42 В, а для роботи у приміщеннях з підвищеною і особливою небезпекою, в тісноті, незручному положенні, коли є небезпека дотику працюючого до металевих, добре заземлених частин, застосовують переносні світильники місцевого освітлення напругою 12 В. У приміщеннях, на робочих місцях, де за умовами безпеки праці, електроприймачі не можуть живитися безпосередньо від мережі напругою до 1000 В, треба застосовувати розподільні або знижувальні трансформатори із вторинною напругою 42 В і нижче.

Захисне відімкнення використовується у тих випадках, коли інші захисні заходи (заземлення, занулення) ненадійні, їх важко здійснити (в умовах вічної мерзлоти), багато коштують або у випадках, коли до безпеки обслуговування пред'являються підвищенні вимоги (в шахтах, кар'єрах), а також у пересувних електроустановках. Найбільше розповсюдження устрої захисного відімкнення знайшли в мережах до 1000 В. Захисне відімкнення обов'язкове для ручних електроінструментів.

Основні вимоги, яким повинні відповідати устрої захисного відімкнення: висока чутливість, малий час відімкнення, селективність дії, здатність здійснювати самоконтроль справності, надійність.

Для захисту від дотику до частин, що знаходяться під напругою, використовується також подвійна ізоляція - електрична ізоляція, що

складається з робочої та додаткової ізоляції. Робоча ізоляція - ізоляція струмоведучих частин електроустановки. Додаткова ізоляція найбільш просто здійснюється виготовленням корпусу з ізоляційного матеріалу (електропобутові прилади).

Відповідальні за пожежну безпеку особи були назначені завідувачем кафедри, відповідальні особи в свою чергу слідкують за приміщенням, станом технологічного та інженерного устаткування, а також за утримання і експлуатацію технічних засобів протипожежного захисту. Запроваджено загально об'єктову інструкцію про заходи пожежної безпеки й відповідні інструкції для всіх пожежне не безпечних приміщень, а також ознайомлено з цією інструкцією всіх працівників, складені плани (схеми) евакуації людей у разі пожежі, затверджено порядок (систему) сповіщення людей про пожежу, визначено категорії будівель і приміщень за вибухопожежною та пожежною небезпекою відповідно до вимог чинних нормативних документів, а також визначено класи зон за "Правилами будови електроустановок". На території встановлено відповідні знаки пожежної безпеки, таблички з вказівкою номеру телефону та порядку виклику пожежної охорони.

Отже, правильний підхід до організації охорони праці на підприємстві, грамотне використання різних нематеріальних способів стимулювання працівників дають останнім необхідне почуття надійності, стабільності й зацікавленості керівництва у своїх співробітниках. Таким чином, завдяки налагодженій охороні праці знижується також плінність кадрів, що в свою чергу благотворно впливає на стабільність усього підприємства.

Вивчення й вирішення проблем, пов'язаних із забезпеченням здорових і безпечних умов, у яких відбувається праця людини – одне з найбільш важливих завдань у розробці нових технологій і систем виробництва. Дослідження й виявлення можливих причин виробничих нещасних випадків,

професійних захворювань, аварій, вибухів, пожеж, і розробка заходів і вимог, спрямованих на усунення цих причин дозволяють створити безпечні й сприятливі умови для праці людини. Комфортні й безпечні умови праці – один з основних факторів, який впливає на продуктивність і безпеку праці, здоров'я працівників.

Електробезпека

Категорії приміщень з електробезпеки поділяються на 3 класи. Тістоприготувальне відділення відноситься до класу підвищеної безпеки. Для запобігання ураження працюючих електричним струмом все обладнання має заземлення. Також є система аварійного відключення живлення при перевантаженні електричної мережі. Потужність електричної мережі розрахована з урахуванням потужності всіх електроприймачів цеху.

Використовуються електромеханічне блокування, яке забезпечує відключення електричного живлення струмоведучих частин при відкриванні доступу до них. Велика увага приділяється захисту складів безтарного зберігання борошна від статичної електрики. Для цього проводяться наступні заходи:

- металеві пневмоходи з'єднуються із заземленими пристроями;
- заземлюються силоси, просіювачі, дозатори борошна, шнеки, розподілювачі та інші пристрої;
- паралельно розташовані трубопроводи з'єднуються між собою для вирівнювання потенціалу статичної електрики струмопровідними перемикачами через кожні 25 см – це попереджує виникнення іскрових електричних розрядів;
- вся електроарматура виконується у вибухонебезпечному виконанні.

Взагалі, вся електроапаратура на заводі відповідає умовам навколишнього середовища в приміщеннях, де вона розташована. Так в пічному відділенні, де відбувається значне виділення тепла, вся електропроводка має ізоляцію, яка

має високу температуру плавлення, а в заварочному – підвищену вологостійкість.

Всі електропроводи і електричні кабелі надійно захищають від пошкоджень захисним покриттям, а саме: електричні кабелі розташовують у металевих трубах відповідного діаметру, а бо у гнучких металевих рукавах, а електродвигуни мають металеве огороження.

Для пуску електродвигунів використовується тільки рубильники закритого типу.

Робоче місце користувача ПК повинне займати площу не меншу за за 6 , висота приміщення повинна бути не менше 3 м. У зв'язку з цим запропоновано організувати робоче місце користувача ПК наступним чином. Висота над рівнем підлоги робочої поверхні, на якій працює користувач, повинна складати 720 мм. Бажано, щоб робочий стіл при необхідності можна було регулювати по висоті в межах 680-780 мм. Оптимальні розміри поверхні столу 1600 x 1000 кв. мм. Під столом повинен бути простір для ніг з розмірами по глибині 650 мм.

Розміщення клавіатури повинно бути у межах 300 мм від краю столу і не більше цього, що забезпечить операторові зручну опору для передпліч.

Відстань між очима користувача й екраном відеодисплея повинна складати 40 - 80 см. Для очей також періодично виконується комплекс вправ для очей. До таких вправ можна віднести наступний комплекс:

1. Часто-часто моргати очима одну-дві хвилини.
2. Прикласти долоні до чола і зробити кілька рухів вниз до підборіддя. Проробити цю вправу 10 разів. Закінчивши її, не поспішати відкривати очі. Сидячи з закритими очима, зосередити свій внутрішній погляд на очних яблуках, на живильних їх судинах. Відкривати очі повільно.
3. Гарненько розігріти долоні, потерши їх одна об одну, скласти їх човниками і потримати напроти очей, не торкаючись до повік, наче прогріваючи очі теплом долонь.

4. При зімкнутих повіках обернути очима за годинниковою стрілкою 20 разів, стільки ж - проти годинникової; потім — 20 разів по горизонталі і стільки ж по вертикалі.

5. Повторити попередню вправу, але вже з відкритими очима.

6. Сфокусувати погляд на кінчику носа, а потім перевести його на якийнебудь об'єкт, що знаходиться в двох-трьох метрах від вас, потім — на дальній об'єкт десь у лінії горизонту. Повторити 20 разів.

7. Подушечками трьох пальців — вказівного, середнього та безіменного — дуже-дуже легко натискати на закриті повіки 8-10 разів.

Виконувати відразу двома руками. Вимоги до робочого стільця користувача наступні: стілець повинен бути обладнаний підйомно-поворотним механізмом, висота сидіння повинна регулюватися в межах 400 - 500 мм, а глибина сидіння повинна складати не менш 380 мм, ширина – не менш 400 мм. Висота опорної поверхні спинки не менше 300 мм, ширина – не менше 380 мм.

Розглянуті можливі небезпеки та шкідливі чинники, які можуть виникнути під час роботи над розробкою підсистеми.

Виявлено, що приміщення відповідає санітарним нормам – мінімальна площа та об'єм з розрахунку на одну людину повністю задовольняють нормативним значенням, пожежні умови також вимагають вимогам.

Рівні шуму, вібрації та загазованості не перевищують нормативних обмежень.

Для підтримання параметрів мікроклімату в приміщенні розташовано кондиціонер та систему водяного опалення для холодної пори року.

Умови праці відповідають вимогам, але рівень шуму наближається до граничного значення. Для зменшення шумового навантаження рекомендується встановити шумоізолюючі вікна та корпуси персональних комп'ютерів.

ВИСНОВОК

У дипломній роботі було описано характеристики та особливості діяльності Кондитерського Дому «ВАЦАК» та його організаційну структуру. Було проведено дослідження діяльності відділу збуту Кондитерського Дому «ВАЦАК» які показали, що потрібно впроваджувати нову систему підтримки його роботи, а також проводити оптимізацію робочих процесів.

В дипломній роботі було розроблено інформаційну систему підтримки діяльності відділу збуту Кондитерського Дому «ВАЦАК». При роботі було використано такі засоби, як:

All Fusion Process Modeler;

All Fusion ERwin Data Modeler;

Microsoft Visual Studio 2015

Microsoft SQL 2016.

Було розроблено моделі ERWin, логічну та фізичну моделі.

Розроблено базу даних за допомогою Microsoft SQL 2016, а також інтерфейс програми за допомогою Microsoft Visual Studio 2015. Визначені основні функції покладені на систему. Проведено опис та порівняння систем аналогів і обґрунтували доцільність створення системи.

Розроблено функціональну модель, здійснено вибір програмного забезпечення, наведено алгоритми реалізації функцій, розроблена інструкція користувача. Описано заходи з охорони праці. Проведений розрахунок економічної ефективності від впровадження системи на підприємстві.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Методичні вказівки до виконання кваліфікаційної бакалаврської роботи для студентів за напрямом підготовки 6.050101 «Комп'ютерні науки» денної та заочної форм навчання / Уклад.: В.В. Самсонов, Л.Ю. Маноха, Т.М. Горлова, Л.Г. Загоровська, О.М. М'якшило, О.А Хлобистова. – К.: НУХТ, 2011. – 15с.;
2. <https://vatsak.com.ua/about/>
3. https://uk.wikipedia.org/wiki/Кондитерський_Дім_«Вацак»
4. https://knowledge.allbest.ru/programming/3c0b65635a3ac68b4c43b88421216c36_0.html
5. http://ito.vspu.net/ENK/2016-2017/Hordiychyk_Case_2016/r/proekt_case_ilyina/10_1.html
6. <http://easy-code.com.ua/2011/07/allfusion-process-modeler-providnij-instrument-vizualnogo-modelyuvannya-biznes-procesiv/>
7. https://informatics.in.ua/programming_csharp/part_01.php
8. <https://docs.microsoft.com/ru-ru/dotnet/framework/get-started/overview>
9. <https://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/>
10. <https://uk.wikipedia.org/wiki/MySQL>
11. https://ru.wikipedia.org/wiki/ERwin_Data_Modeler
12. <https://erwin.com/products/erwin-data-modeler/>
13. <http://um.co.ua/2/2-15/2-15558.html>
14. https://ru.wikipedia.org/wiki/Microsoft_SQL_Server
15. ГОСТ 12.4.009-83 «Пожарная техника для защиты объектов. Основные виды. Размещение и обслуживание», 1983. – 15 с.
16. ДСТУ 2120-93 Хлібопекарське виробництво. Терміни та визначення.
17. ДСТУ 2887-94 Пакування та маркування. Терміни та визначення.

ERwin Process Modeler - [(2 13) Кондитерський Дім "ВАЦАК" - 3lev [3lev.bp1]]

File Edit View Diagram Dictionary Model Services Tools Window Help

69%

3lev

- Context
- Swim Lane diagrams
- Organization Charts
- Node Tree diagrams
- FEO diagrams
- IDEF3 Scenario diagrams**
 - Кондитерський Дім "ВАЦАК"

USED AT:	AUTHOR:	DATE: 26.01.2021	WORKING	READER	DATE	CONTEXT:
	PROJECT: 3lev	REV: 26.01.2021	DRAFT			TOP
			RECOMMENDED			
			PUBLICATION			

NOTES: 1 2 3 4 5 6 7 8 9 10

```

graph TD
    Input1[Перелік дистрибуторів] --> Node1[Регістрація дистрибуторів в системі]
    Input2[Продукція] --> Node2[Формування договору на збут]
    Input3[План реалізації продукції] --> Node2
    Input4[Нормативно правові документи] --> Node2
    Node1 --> Node2
    Node2 --> Node3[Відправка продукції]
    Node2 --> Node4[Формування звітів про роботу]
    Node3 --> Node4
    Node4 --> Output1[Договори з дистрибуторами]
    Node4 --> Output2[Звіт про збут продукції]
    Node4 --> Output3[Документація та відвантаження]
    
```

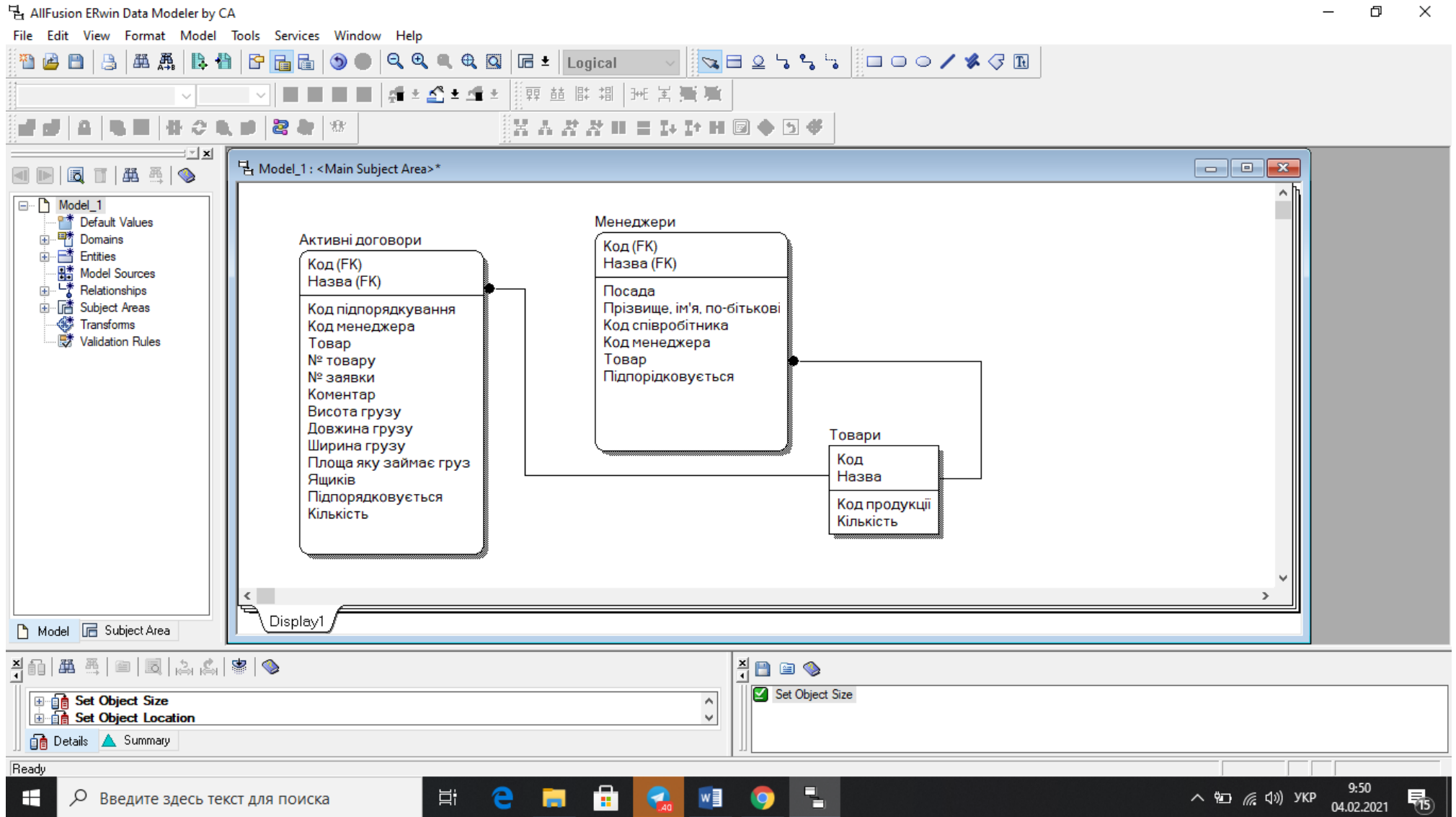
PK Менеджер відділу

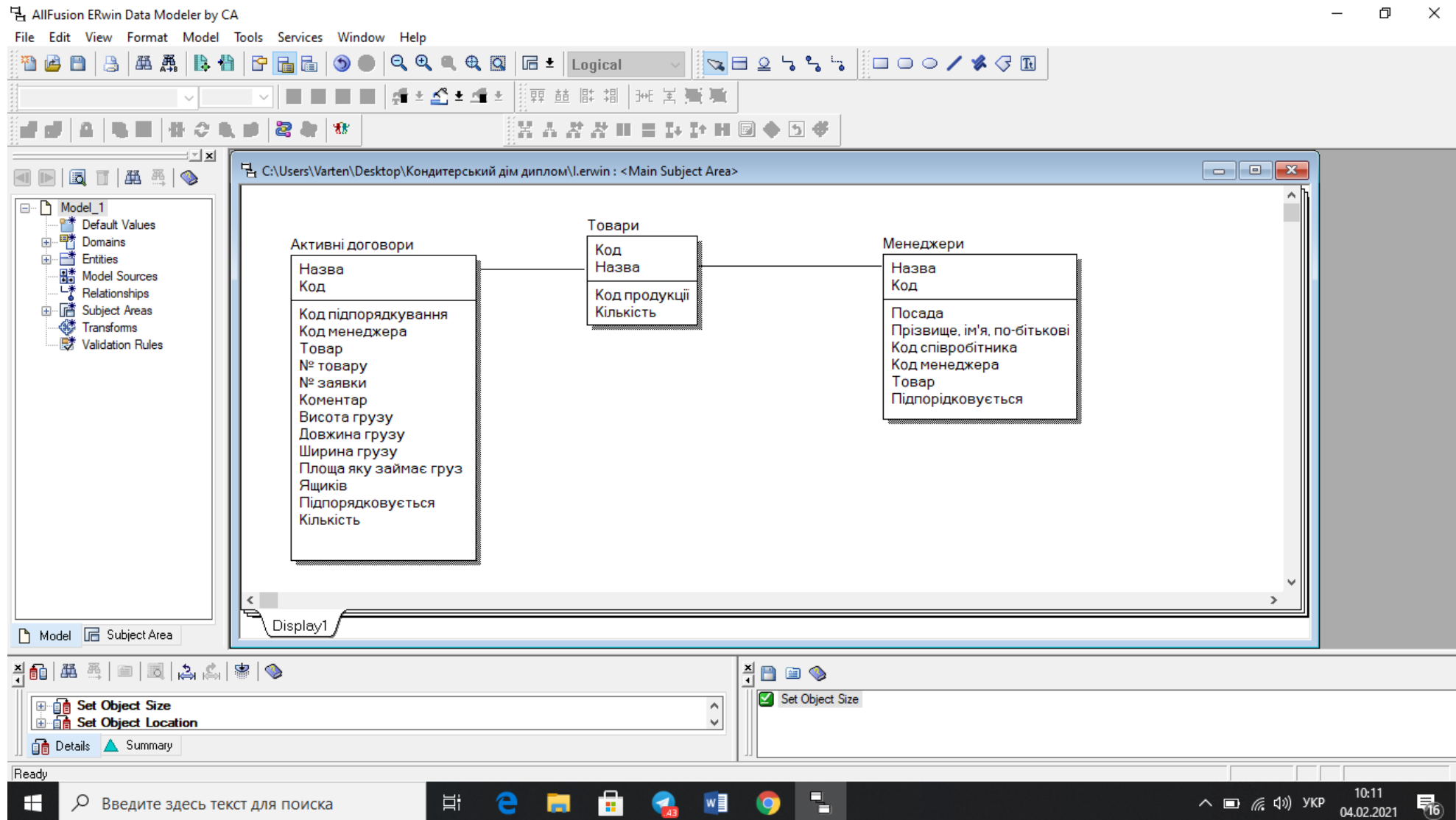
NODE:	TITLE:	NUMBER:
2	Είραεοαοηυεεε Α*1 "ΑΑΟΑΕ"	

Activities Diagrams Objects

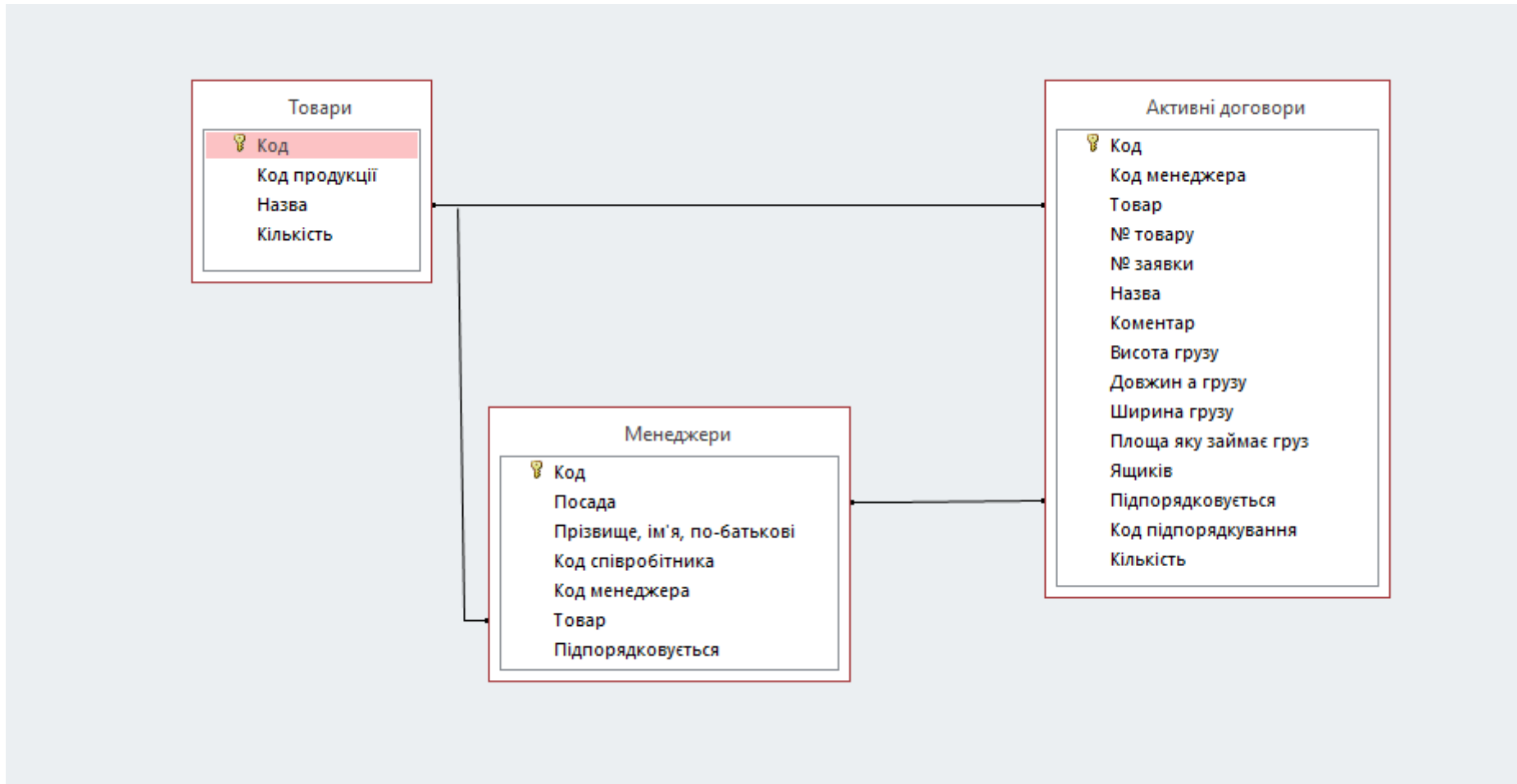
Введіть текст для пошука

23:33 26.01.2021





Додаток Г. Згенерована база даних



Додаток Д. Фрагмент лістингу програми

Головна форма

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

namespace AIS
{
    public partial class Main_form : Form
    {
        db_control dbc;
        public Main_form()
        {
            InitializeComponent();
            dbc = new db_control(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\Db_AIS.mdf;Integrated
Security=True");
            //tabControl1.Visible = false;
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            dbc.con.Open();
            dbc.FillDB("Units", unitsDataGridView);
            unitsDataGridView.Columns[0].Visible = false;
//+++++
            unitsDataGridView.Columns[4].Visible = false;
//+++++
            dbc.FillDB("Housing", housingDataGridview1);
            housingDataGridview1.Columns[0].Visible = false; //-----
            dbc.FillDB("Rooms", roomsDataGridview);
            roomsDataGridview.Columns[1].Visible = false; ////////////////
            roomsDataGridview.Columns[0].Visible = false; ////////////////
            roomsDataGridview.Columns[6].Visible = false; ////////////////
        }
    }
}
```

```

roomsDataGridView.Columns[7].Visible = false; ///////////////////////////////////////////////////////////////////
roomsDataGridView.Columns[8].Visible = false; ///////////////////////////////////////////////////////////////////
roomsDataGridView.Columns[9].Visible = false; ///////////////////////////////////////////////////////////////////
roomsDataGridView.Columns[13].Visible = false; ///////////////////////////////////////////////////////////////////
roomsDataGridView.Columns[14].Visible = false; ///////////////////////////////////////////////////////////////////

}

public void vis()
{
    tabControl1.Visible = true;
}

private void button1_Click(object sender, EventArgs e)
{
    dbc.AddWindowUnits();
    dbc.examp1.Text = "Додати менеджера";
}

private void button2_Click(object sender, EventArgs e)
{
    try
    {
        dbc.DeleteCurrentRowDB("Units", unitsDataGridView,
Convert.ToInt32(unitsDataGridView.CurrentRow.Cells[0].Value), true);
        unitsDataGridView.Columns[0].Visible = false;
//+++++
        unitsDataGridView.Columns[4].Visible = false;
//+++++
        richTextBox_units.Text = "";
    }
    catch
    {
        MessageBox.Show("Не обрано запис для редагування!", "Помилка",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

```

```

    }
}

private void button3_click(object sender, EventArgs e)
{
    dbc.UpdateTable("Units", unitsDataGridView,true);
}

private void Main_form_Activated(object sender, EventArgs e)
{
    dbc.FillDB("Units", unitsDataGridView);
    unitsDataGridView.Columns[0].Visible = false;
//+++++
    unitsDataGridView.Columns[4].Visible = false;
//+++++
    dbc.FillDB("Housing", housingDataGridView1);
    housingDataGridView1.Columns[0].Visible = false; //-----
    dbc.FillDB("Rooms", roomsDataGridView);
    roomsDataGridView.Columns[1].Visible = false; ///////////////////////////////////////////////////////////////////
    roomsDataGridView.Columns[0].Visible = false; ///////////////////////////////////////////////////////////////////
    roomsDataGridView.Columns[6].Visible = false; ///////////////////////////////////////////////////////////////////
    roomsDataGridView.Columns[7].Visible = false; ///////////////////////////////////////////////////////////////////
    roomsDataGridView.Columns[8].Visible = false; ///////////////////////////////////////////////////////////////////
    roomsDataGridView.Columns[9].Visible = false; ///////////////////////////////////////////////////////////////////
    roomsDataGridView.Columns[13].Visible = false; ///////////////////////////////////////////////////////////////////
    roomsDataGridView.Columns[14].Visible = false; ///////////////////////////////////////////////////////////////////
}

private void Add_btn_housing_Click(object sender, EventArgs e)
{
    dbc.AddWindowHousing();
    dbc.examp2.Text = "Додавання товарів";
}

private void Delete_btn_housing_Click(object sender, EventArgs e)

```

```

    {
        try
        {
            dbc.DeleteCurrentRowDB("Housing", housingDataGridView1,
Convert.ToInt32(housingDataGridView1.CurrentRow.Cells[0].Value), true);
            housingDataGridView1.Columns[0].Visible = false;//-----
--
            richTextBox_housing.Text = "";
        }
        catch
        {
            MessageBox.Show("Не обрано запис для редагування!", "Помилка",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }

private void Update_btn_housing_Click(object sender, EventArgs e)
{
    dbc.UpdateTable("Housing", housingDataGridView1,true);
}

private void Add_btn_rooms_Click(object sender, EventArgs e)
{
    dbc.AddWindowRooms();
    dbc.examp3.Text = "Додати активний договір";
    dbc.FillDB("Rooms", roomsDataGridView);
    roomsDataGridView.Columns[0].Visible = false; //////////////////////////////////
    roomsDataGridView.Columns[1].Visible = false; //////////////////////////////////
    roomsDataGridView.Columns[6].Visible = false; //////////////////////////////////
    roomsDataGridView.Columns[7].Visible = false; //////////////////////////////////
    roomsDataGridView.Columns[8].Visible = false; //////////////////////////////////
    roomsDataGridView.Columns[9].Visible = false; //////////////////////////////////
    roomsDataGridView.Columns[13].Visible = false; //////////////////////////////////
    roomsDataGridView.Columns[14].Visible = false; //////////////////////////////////
}

```

```

}

private void Delete_btn_rooms_Click(object sender, EventArgs e)
{
    try
    {
        dbc.DeleteCurrentRowDB("Rooms", roomsDataGridView,
Convert.ToInt32(roomsDataGridView.CurrentRow.Cells[0].Value), true);
        roomsDataGridView.Columns[0].Visible = false; //////////////////////////////////////
        roomsDataGridView.Columns[1].Visible = false; //////////////////////////////////////
        roomsDataGridView.Columns[6].Visible = false; //////////////////////////////////////
        roomsDataGridView.Columns[7].Visible = false; //////////////////////////////////////
        roomsDataGridView.Columns[8].Visible = false; //////////////////////////////////////
        roomsDataGridView.Columns[9].Visible = false; //////////////////////////////////////
        roomsDataGridView.Columns[13].Visible = false; //////////////////////////////////////
        roomsDataGridView.Columns[14].Visible = false; //////////////////////////////////////
        richTextBox_rooms.Text = "";
    }
    catch
    {
        MessageBox.Show("Не обрано запис для видалення!", "Помилка",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

private void Update_btn_rooms_Click(object sender, EventArgs e)
{
    dbc.UpdateTable("Rooms", roomsDataGridView,true);
}

private void edit_btn_Click(object sender, EventArgs e)
{
    try
    {
        dbc.AddWindowUnits();
    }
}

```

```

        dbc.examp1.Text = "Редагувати інформацію про менеджера";
        dbc.SelectEditRow("Units", (int)unitsDataGridView.CurrentRow.Cells[0].Value);
        dbc.examp1.edit = true;
        dbc.examp1.temp_id = Convert.ToInt32(dbc.ds.Tables[0].Rows[0].ItemArray[0]);
        dbc.examp1.nameTextBox.Text =
dbc.ds.Tables[0].Rows[0].ItemArray[1].ToString();
        dbc.examp1.full_nameTextBox.Text =
dbc.ds.Tables[0].Rows[0].ItemArray[2].ToString();
        dbc.examp1.resp_personTextBox.Text =
dbc.ds.Tables[0].Rows[0].ItemArray[3].ToString();
        dbc.examp1.comboBox1.Text =
dbc.ds.Tables[0].Rows[0].ItemArray[5].ToString();
        dbc.examp1.comboBox_dependence.Text =
dbc.ds.Tables[0].Rows[0].ItemArray[6].ToString();
    }
    catch
    {
        dbc.examp1.Close();
        MessageBox.Show("Не обрано запис для редагування!", "Помилка",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

private void unitsDataGridView_Click(object sender, EventArgs e)
{
    try
    {
        richTextBox_units.Text =
dbc.Select_units(unitsDataGridView.CurrentRow.Cells[1].Value.ToString());
    }
    catch { }
}

private void housingDataGridView1_Click(object sender, EventArgs e)
{

```

```

        try
        {
            richTextBox_housing.Text =
dbc.Select_housing(Convert.ToInt32(housingDataGridview1.CurrentRow.Cells[0].Value));
        }
        catch { }
    }

private void roomsDataGridview_Click(object sender, EventArgs e)
{
    try
    {
        richTextBox_rooms.Text =
dbc.Select_rooms(Convert.ToInt32(roomsDataGridview.CurrentRow.Cells[0].Value),true);
    }
    catch { }
}

private void button1_Click_1(object sender, EventArgs e)
{
    try
    {
        dbc.AddWindowHousing();
        dbc.examp2.Text = "Редагування товару";
        dbc.SelectEditRow("Housing",
(int)housingDataGridview1.CurrentRow.Cells[0].Value);
        dbc.examp2.edit = true;
        dbc.examp2.temp_id = Convert.ToInt32(dbc.ds.Tables[0].Rows[0].ItemArray[0]);
        dbc.examp2.textBox_corp_num.Text =
dbc.ds.Tables[0].Rows[0].ItemArray[1].ToString();
        dbc.examp2.nameTextBox1.Text =
dbc.ds.Tables[0].Rows[0].ItemArray[2].ToString();
        dbc.examp2.floor_countTextBox.Text =
dbc.ds.Tables[0].Rows[0].ItemArray[3].ToString();
    }
    catch { }
}

```

```

    }
    catch
    {
        dbc.examp2.Close();
        MessageBox.Show("Не выбрана запись для редактирования!", "Ошибка",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

private void button2_Click_1(object sender, EventArgs e)
{
    try
    {
        dbc.AddWindowRooms();
        dbc.examp3.Text = " Редагувати догорів";
        dbc.SelectEditRow("Rooms",
(int)roomsDataGridView.CurrentRow.Cells[0].Value);
        dbc.examp3.edit = true;
        dbc.examp3.temp_id = Convert.ToInt32(dbc.ds.Tables[0].Rows[0].ItemArray[0]);
        dbc.examp3.comboBox_houisng.Text =
dbc.ds.Tables[0].Rows[0].ItemArray[2].ToString();
        dbc.examp3.TextBox_aud_num.Text =
dbc.ds.Tables[0].Rows[0].ItemArray[3].ToString();
        dbc.examp3.TextBox_pom_num.Text =
dbc.ds.Tables[0].Rows[0].ItemArray[4].ToString();
        dbc.examp3.room_nameTextBox.Text =
dbc.ds.Tables[0].Rows[0].ItemArray[5].ToString();
        dbc.examp3.richTextBox_room_equipment.Text =
dbc.ds.Tables[0].Rows[0].ItemArray[6].ToString();
        dbc.examp3.room_heightTextBox.Text =
dbc.ds.Tables[0].Rows[0].ItemArray[7].ToString();
        dbc.examp3.room_lenghtTextBox.Text =
dbc.ds.Tables[0].Rows[0].ItemArray[8].ToString();
        dbc.examp3.room_widthTextBox.Text =
dbc.ds.Tables[0].Rows[0].ItemArray[9].ToString();
    }
}

```

```

        dbc.examp3.textBox_square.Text =
dbc.ds.Tables[0].Rows[0].ItemArray[10].ToString();
        dbc.examp3.textBox_seats.Text =
dbc.ds.Tables[0].Rows[0].ItemArray[11].ToString();
        dbc.examp3.comboBox_secured.Text =
dbc.ds.Tables[0].Rows[0].ItemArray[12].ToString();
        dbc.examp3.comboBox_roomfloor.Text =
Convert.ToInt32(dbc.ds.Tables[0].Rows[0].ItemArray[14]).ToString();
    }
    catch
    {
        dbc.examp3.Close();
        MessageBox.Show("Не обрано запис для редагування!", "Помилка",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

private void button3_Click_1(object sender, EventArgs e)
{
    for (int i = 0; i < roomsDataGridView.RowCount; i++)
    {
        roomsDataGridView.Rows[i].Selected = false;

        if (roomsDataGridView.Rows[i].Cells[3].Value.ToString() ==
textBox1.Text.ToString())
        {
            roomsDataGridView.Rows[i].Selected = true;
            break;
        }
    }
}

private void label2_Click(object sender, EventArgs e)

```

```

    {

    }

    private void label1_Click(object sender, EventArgs e)
    {

    }

    private void textBox3_TextChanged(object sender, EventArgs e)
    {

    }

    private void textBox2_TextChanged(object sender, EventArgs e)
    {

    }

    private void button4_Click(object sender, EventArgs e)
    {
        if (textBox2.Text == "admin" && textBox3.Text == "admin") panel2.Visible = false;
    }
}

```

Клас доступу до БД

```

using System;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;

namespace AIS
{

```

```

class db_control
{
public db_control(string path_to_bd)
{
    con = new SqlConnection(path_to_bd);
}
public SqlConnection con;
public Add_row_form examp1;
public Add_Form_Housing examp2;
public Add_Form_Rooms examp3;
public DataSet ds;
public SqlDataAdapter adap;
public void AddWindowUnits()
{
    examp1 = new Add_row_form();
    examp1.Show();
}
public void AddWindowHousing()
{
    examp2 = new Add_Form_Housing();
    examp2.Show();
}
public void AddWindowRooms()
{
    examp3 = new Add_Form_Rooms();
    examp3.Show();
}

public void FillDB(string t_n, DataGridView dgv)
{
    string table_name = t_n;
    DataGridView datagridview = dgv;
    datagridview.DataSource = null;
    adap = new SqlDataAdapter("SELECT * FROM " + table_name + " WHERE [Код] >
1", con);

```

```

ds = new DataSet();
adap.Fill(ds, table_name);
datagridview.DataSource = ds.Tables[0];
}
public void DeleteCurrentRowDB(string t_n, DataGridView dgv, int id_del, bool more)
{
    if (more)
    {
        DialogResult result;
        result = MessageBox.Show("Ви дійсно хочете видалити запис?", "Видалити запис", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

        if (result == DialogResult.Yes)
        {
            string table_name = t_n;
            DataGridView datagridview = dgv;
            int id = id_del;
            try
            {
                SqlCommand command = new SqlCommand("DELETE FROM " +
table_name + " WHERE [Код] = " + id, con);
                command.ExecuteNonQuery();

                FillDB(table_name, datagridview);
            }
            catch
            {
                result = MessageBox.Show("В запису є пов'язані записи, видалити всі разом?", "Помилка", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

                if (result == DialogResult.Yes)
                {
                    SqlCommand command = new SqlCommand("DELETE FROM Rooms
WHERE [Код менеджера] = " + id, con);
                    command.ExecuteNonQuery();
                }
            }
        }
    }
}

```



```

    }
}
}
public void UpdateTable(string t_n, DataGridView dgv,bool more)
{
    try
    {
        string table_name = t_n;
        DataGridView datagridview = dgv;
        adap = new SqlDataAdapter("SELECT * FROM " + table_name, con);
        ds = new DataSet();
        adap.Fill(ds, table_name);
        adap.Update(ds, table_name);
        FillDB(table_name, datagridview);
        if (more)
            MessageBox.Show("Інформацію оновлено", "Оновлення",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Помилка", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    }

}

public DataSet SelectEditRow(string t_n, int id)
{
    string table_name = t_n;
    int id_row = id;
    adap = new SqlDataAdapter("SELECT * FROM " + table_name + " WHERE [Код] =
" + id_row, con);
    ds = new DataSet();
    adap.Fill(ds, table_name);
    return ds;
}
}

```

```

public string Select_room_and_housing(int id)
{
    int count_courpuses = 0;
    string g = "";
    adap = new SqlDataAdapter("SELECT * FROM Units where [Посада] = " + id, con);
    ds = new DataSet();
    adap.Fill(ds, "Housing");
    count_courpuses = ds.Tables[0].Rows.Count;
    if (count_courpuses >= 1)
    {
        foreach (DataRow row in ds.Tables[0].Rows)
            g = Select_rooms(id, false);
        g += "\n" + "Кількість товарів = " + count_courpuses;
    }
    else
        g = "кількість товарів = 0";
    return g;
}

public string Select_rooms(int id, bool more)
{
    string g = "";
    try
    {
        double S = 0, V = 0;
        if (more)
            adap = new SqlDataAdapter("SELECT [Довжина грузу],[Ширина
грузу],[Висота грузу],[Кількість],[Коментар] FROM Rooms where [Код] = " + id, con);
        else
            adap = new SqlDataAdapter("SELECT [Довжина грузу],[Ширина
грузу],[Висота грузу],[Кількість] FROM Rooms where [Код] = " + id, con);
        ds = new DataSet();
        adap.Fill(ds, "Housing");
        foreach (DataRow row in ds.Tables[0].Rows)
            {

```

```

        S += Convert.ToDouble(row.ItemArray[0]) *
Convert.ToDouble(row.ItemArray[1]);
        V += Convert.ToDouble(row.ItemArray[0]) *
Convert.ToDouble(row.ItemArray[0]) * Convert.ToDouble(row.ItemArray[2]);
    }
    g = "Загальна площа грузу = " + S + " кв.м" +
    '\n' + "Загальний об'єм грузу = " + V + " кв.м" + '\n' + '\n';
    if (ds.Tables[0].Rows[0].ItemArray[4].ToString().Length > 0)
        g += "Коментар до договору:" + '\n' + ds.Tables[0].Rows[0].ItemArray[4] + '\n'
+ '\n';

    else
        g += "Коментарі до до договору:" + '\n' + "Відсутні!" + '\n' + '\n';
    g += "Кількість = " + ds.Tables[0].Rows[0].ItemArray[3];
    return g;
}
catch
{
    return g = "";
}
}
public string Select_housing(int id)
{
    string g = "";
    try
    {
        int count_facul = 0, count_pom = 0;
        string name_facul = "";
        adap = new SqlDataAdapter("SELECT [Підпорядковується] FROM Units where
[Код менеджера] = " + id, con);
        ds = new DataSet();
        adap.Fill(ds, "Units");
        foreach (DataRow row in ds.Tables[0].Rows)
        {
            count_facul++;
            name_facul +=row.ItemArray[0].ToString() + '\n';

```

```

    }
    adap = new SqlDataAdapter("SELECT [Підпорядковується] FROM Rooms where
[Код менеджера] = " + id, con);
    ds = new DataSet();
    adap.Fill(ds, "Rooms");
    foreach (DataRow row in ds.Tables[0].Rows)
        count_pom++;
    if (count_facul == 0)
        g += "Керівники продаж: відсутні";
    else
        g += "Керівники продаж: " + '\n' + name_facul + '\n' +
            "Кількість менеджерів: " + count_facul + '\n' +
            "Кількість договорів: " + count_pom;
    return g;
}
catch
{
    return g = "Реалізатори та договори відсутні!";
}
}
public string Select_units(string str)
{
    string g = "";
    try
    {
        int count_pom = 0;
        float s_pom = 0;
        adap = new SqlDataAdapter("select [Площа яку займає груз] FROM Rooms
WHERE [Підпорядковується] = N" + str + "'", con);
        ds = new DataSet();
        adap.Fill(ds, "Rooms");
        foreach (DataRow row in ds.Tables[0].Rows)
        {
            count_pom++;
            s_pom += Convert.ToSingle(row.ItemArray[0]);

```

```
    }  
    if (count_rom == 0)  
        g += "Загальна площа грузу: відсутня";  
    else  
        g += "Загальна площа: " + s_rom + " кв.м" + '\n' +  
            "Кількість договорів: " + count_rom;  
    return g;  
    }  
    catch  
    {  
        return g = "Інформація відсутня!";  
    }  
    }  
    }  
    }
```