

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ

к.т.н, доц. Михелєв І.Л.

" ____ " _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "магістр"

на тему:

Дослідження технологій та розробка інформаційної системи для складського
господарства пивзаводу

Виконав: студент групи 6142м

Залітко А.С. Прізвище І.Пб.

(підпис)

Керівник роботи:

(посада, науковий ступінь вчене звання)

Казимиренко Ю.О. Прізвище І.Пб.

(підпис)

м. Миколаїв – 2023 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами

Кафедра інформаційних управляючих систем та технологій

Спеціальність 122 "Комп'ютерні науки"

Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми

к.т.н, доц. Ажищев В.Ф.

" ____ " _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти "магістр"

Студенту Залітко Антону Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження технологій та розробка інформаційної системи для складського господарства пивзаводу

Керівник роботи

Казимиренко Юлія Олексіївна

Затверджені наказом ректора № 1098-уч. від "18" вересня 2023 року.

2. Термін подання роботи: 15 грудня 2023 р.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем, матеріали наукового стажування

4. Перелік питань, що належать до розробки (найменування розділів)

1.сучасний стан інформаційної підтримки складського господарства пивзаводу;
2.проектування складського господарства пивзаводу; 3. Реалізація ІС складського господарства пивзаводу; 4. Рорахунок економічної ефективності впровадження ІС складського господарства пивзаводу; 5. Охорона праці; 6. Охорона навколишнього середовища

5. Перелік презентаційних матеріалів _____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Казимиренко Ю.О.		
2			
3			
4			

7. Дата видачі завдання 05 вересня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Вивчення і опис предметної області	01.11.2023	
2	Розробка концепції системи	14.11.2023	
3	Проектування системи	16.11.2023	
4	Реалізація проекту	21.11.2023	
5	Вирішення питань охорони праці	26.11.2023	
6	Виконання графічних документів	01.12.2023	
7	Оформлення пояснювальної записки	08.12.2023	
8	Подання дипломної роботи (ДР) на кафедральний захист	10.12.2023	
9	Подання ДР рецензенту	11.12.2023	
10	Подання ДР на державний захист	13.12.2023	

Студент

(підпис)

Залітко А.С.

(ПІБ)

Керівник роботи

(підпис)

Казимиренко Ю.О.

(ПІБ)

АНОТАЦІЯ

Магістерська робота «Дослідження технологій та розробка інформаційної системи для складського господарства пивзаводу» за спеціальністю 122 – комп'ютерні науки.

Робота присвячена вирішенню питання розробки та впровадження інформаційної системи для управління складським господарством пивзаводу.

Метою роботи є дослідження технологій та розробка інформаційної системи для управління процесами складського господарства пивзаводу, впровадження якої спрямовано на систематизацію обліку продукції, ресурсів та скорочення часу на обробку інформації.

Наукова новизна роботи полягає в наступному:

- вдосконалено підхід до управління процес обробки і систематизації даних про продукцію, сировину, споживачів, документацію, що на відміну від існуючих, дозволяє підвищити ефективність виконання розв'язання логістичних задач.
- вперше для складського господарства пивзаводу запропоновано теоретично-практичне підґрунтя координації цілей з розширенням можливостей та виходом на міжнародний ринок в умовах воєнного стану.

Наукове значення роботи полягає в отриманні результатів, які будуть спрямовані на вирішення проблем інформаційного забезпечення при управлінні складським господарством на харчовому підприємстві в умовах воєнного стану.

Практичне значення роботи полягає в розробці заходів з інформаційної підтримки підприємства харчової промисловості в умовах наслідків часткової руйнації та дефіциту прісної води у місті Миколаєві.

Методи дослідження: методи системного аналізу; принципи процедурного програмування, класичні технології проектування баз даних.

Апробація роботи: результати були представлені на міжнародній науково-технічній конференції.

Робота написана на ____ сторінках машинописного тексту, містить ____ рисунків, ____ таблиць, ____ додатки і список використаних джерел з ____ найменувань.

Ключові слова: складське господарство, пивзавод, інформаційна система, система управління складом, база даних.

ABSTRACT

Master's thesis "Research of technologies and development of an information system for warehouse management of a brewery" in specialty 122 - computer science.

The work is devoted to solving the issue of development and implementation of an information system for warehouse management of a brewery. The purpose of the work is the research of technologies and the development of an information system for managing the warehouse management processes of the brewery, the implementation of which is aimed at systematizing the accounting of products, resources and reducing the time for processing information.

The scientific novelty of the work is as follows:

- the approach to managing the process of processing and systematizing data on products, raw materials, consumers, and documentation has been improved, which, unlike the existing ones, allows to increase the efficiency of solving logistical problems.
- for the first time, a theoretical and practical basis for the coordination of goals with the expansion of opportunities and access to the international market in the conditions of martial law was proposed for the warehouse management of the brewery.

The scientific value of the work consists in obtaining results that will be aimed at solving the problems of information support in the management of warehouse management at a food enterprise in the conditions of martial law.

The practical significance of the work consists in the development of information support measures for the food industry enterprise in the conditions of the consequences of partial destruction and shortage of fresh water in the city of Mykolaiv.

Research methods: methods of system analysis; principles of procedural programming, classic database design technologies.

Approbation of the work: the results were presented at an international scientific and technical conference.

The work is written on ____ pages of typewritten text, contains ____ figures, ____ tables, ____ appendices and a list of used sources from ____ titles.

Keywords: warehousing, brewery, information system, warehouse management system, database.

Зміст

1. СУЧАСНИЙ СТАН ІНФОРМАЦІЙНОЇ ПІДТРИМКИ СКЛАДСЬКОГО ГОСПОДАРСТВА ПИВЗАВОДУ	10
1.1 Характеристика складського господарства пивзаводу	10
1.2 Аналіз готових рішень	11
1.3 Організація виробництва	13
1.3.1 Впровадження в експлуатацію.....	13
1.3.2 Сутність, напрямки, цілі і завдання організації праці.....	15
1.3.3 Технолого-нормувальна карта розробки інформаційної системи.....	18
1.3.4 Розрахунок чисельності розробників.....	19
1.4 Аналіз невирішених питань та постановка завдань досліджень	21
2. ПРОЕКТУВАННЯ ІС СКЛАДСЬКОГО ГОСПОДАРСТВА ПИВЗАВОДУ	24
2.1 Теоретичні передумови моделювання інформаційної системи.....	24
2.2 Моделювання та конструювання	25
2.2.1 Розробка моделі системи	25
2.2.2 Дослідження потоків даних.....	27
2.2.3 Концептуальна модель	33
2.2.4 Розробка діаграми станів програми.....	34
2.3 Технічний проект	35
2.3.1 Логічна модель даних	35
2.3.2 Розробка архітектури програми	40
2.3.3 Специфікація модулів	40
2.3.4 Алгоритм роботи програми.....	43
2.4 Робочий проект.....	43
2.4.1 Вибір засобів розробки програмного забезпечення ІС	43
2.4.2 Фізична модель бази даних.....	45
2.4.3 Кодування	48
2.4.4 Тестування та налагодження програми	50
3. РЕАЛІЗАЦІЯ ІС СКЛАДСЬКОГО ГОСПОДАРСТВА ПИВЗАВОДУ	53

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВПРОВАДЖЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ СКЛАДСЬКОГО ГОСПОДАРТСТВА ПІВЗАВОДУ	63
4.1 Загальні положення	63
4.2 Розрахунок вартості матеріально – енергетичних витрат.....	64
4.3 Розрахунок фонду оплати праці за виконану роботу	66
4.4 Розрахунок ціни розробки	68
4.5 Розрахунок економічної ефективності та строку окупності розробленої ІС	71
5 ОХОРОНА ПРАЦІ	73
5.1 Вступ.....	73
5.2 Аналіз шкідливих та небезпечних факторів	73
5.3 Розрахунок системи штучного освітлення приміщення відділу інформаційних технологій	75
5.4 Розробка заходів щодо зменшення впливу шкідливих та небезпечних факторів при роботі з ПК.....	76
5.4.1 Заходи щодо зменшення впливу.....	76
5.4.2 Техніка безпеки при роботі на комп'ютері	77
6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	79
6.1 Вступ.....	79
6.2 Забруднення навколишнього середовища	79
6.3 Розробка заходів щодо зменшення забруднення навколишнього середовища.....	80
ЗАГАЛЬНІ ВИСНОВКИ	83
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	85
Додаток А – код розробки.....	86
Додаток Б – тези	113
Додаток В – Презентаційні матеріали	115

ВСТУП

Важливим проблемним питанням щодо інформаційного забезпечення діяльності підприємств харчової промисловості є створення єдиної інформаційної системи, яка охоплює взаємодію потоків, що відображують приймання, зберігання, комплектацію, відвантаження та облік товарів. Успішне виконання цих процесів обумовлено ефективною дією складської інформаційної системи (ІС).

Сучасні ІС складського управління повинні відповідати високим стандартам, забезпечуючи автоматизацію усіх етапів складської діяльності, контроль за рухом товарів, облік товарних запасів та взаємодію з іншими підсистемами підприємства. Однак для реалізації цих вимог вимагається значний фінансовий та кадровий ресурс, оскільки складські ІС є складними системами, що вимагають спеціалізованої підготовки для розробки та експлуатації.

Актуальність дослідження обумовлена кількома факторами, зокрема ростом важливості складської діяльності в логістичних системах, зростанням вимог до ефективності цих процесів та розвитком інформаційних технологій для автоматизації складських процесів.

Дослідження актуальності теми підтверджує необхідність розробки ефективних ІС для управління складом, які відповідають сучасним стандартам, економічно обґрунтовані та легкі у використанні.

В умовах відродження виробництва під час військового стану, запуск діяльності пивзаводу є важливою складовою відновлення міста Миколаєва.

Об'єктом дослідження є процес обробки і систематизації даних про продукцію, сировину, споживачів, документацію. Методи дослідження включають аналіз, моделювання та економічні розрахунки.

Предмет дослідження – механізми та моделі проєктування інформаційної системи для обслуговування складського господарства пивзаводу

Метою роботи є дослідження технологій та розробка інформаційної системи для управління процесами складського господарства пивзаводу,

впровадження якої спрямовано на систематизацію обліку продукції, ресурсів та скорочення часу на обробку інформації.

Завдання досліджень

1) проаналізувати сучасний стан інформаційного забезпечення складського господарства пивзаводу та наявні потенційні системи, які можна впровадити;

2) дослідити основні принципи проектування інформаційної системи та технології баз даних складського господарства та дослідити їх.

3) розробити програмне забезпечення та розглянути на певному прикладі реалізацію інформаційної системи;

4) надати техніко-економічне обґрунтування щодо доцільності впровадження інформаційної системи для управління складом пивзаводу;

5) розглянути заходи щодо охорони праці та навколишнього середовища при розробці інформаційної системи.

Наукова новизна роботи полягає в наступному:

- вдосконалено підхід до управління процес обробки і систематизації даних про продукцію, сировину, споживачів, документацію, що на відміну від існуючих, дозволяє підвищити ефективність виконання розв'язання логістичних задач.

- вперше для складського господарства пивзаводу запропоновано теоретично-практичне підґрунтя кординації цілей з розширенням можливостей та виходом на міжнародний ринок в умовах воєнного стану.

Наукове значення роботи полягає в отриманні результатів, які будуть спрямовані на вирішення проблем інформаційного забезпечення при управлінні складським господарством на харчовому підприємстві в умовах воєнного стану.

Практичне значення роботи полягає в розробці заходів з інформаційної підтримки підприємства харчової промисловості в умовах наслідків часткової руйнації та дефіциту прісної води у місті Миколаєві.

Апробація результатів досліджень. Основні положення і результати роботи доповідалися на XXXIX-й Міжнародній науково-практичній конференції

«Сучасні аспекти модернізації науки: стан, проблеми, тенденції розвитку» (07 грудня 2023 року, м. Гетеборг, Швеція, дистанційно)

Публікації за результатами магістерської роботи:

Залітко А.С. Особливості впровадження інформаційних систем на харчових підприємствах України / Залітко А.С. // Сучасні аспекти модернізації науки: стан, проблеми, тенденції розвитку: матеріали XXXIX Міжнародної науково-практичної конференції // м. Гетеборг, Швеція: ГО «ВАДНД», 2023 – С. 447-449.

1. СУЧАСНИЙ СТАН ІНФОРМАЦІЙНОЇ ПІДТРИМКИ СКЛАДСЬКОГО ГОСПОДАРСТВА ПИВЗАВОДУ

1.1 Характеристика складського господарства пивзаводу

Складське господарство пивзаводу – це система складів, що забезпечує зберігання сировини, матеріалів, готової продукції та інших матеріальних цінностей, необхідних для виробництва пива[1]. Складське господарство пивзаводу включає в себе:

Склад сировини – призначений для зберігання зерна, солоду, хмелю та інших матеріалів, необхідних для виробництва пива. Склад сировини, як правило, має відповідні умови для зберігання цих матеріалів, наприклад, низьку температуру та вологість.

Склад готової продукції – призначений для зберігання готового пива. Склад готової продукції, як правило, має відповідні умови для зберігання пива, наприклад, темряву та низьку температуру.

Склад допоміжних матеріалів – призначений для зберігання допоміжних матеріалів, що використовуються в процесі виробництва пива, наприклад, тару, обладнання та ін[1].

Складське господарство пивзаводу є важливою частиною виробництва пива. Правильна організація складського господарства забезпечує ефективне зберігання матеріальних цінностей та мінімізує їхні втрати[1].

Ось деякі з основних завдань складського господарства пивзаводу:

- приймання сировини та матеріалів;
- розміщення сировини та матеріалів на складі;
- зберігання сировини та матеріалів у належних умовах;
- підготовка сировини та матеріалів до виробництва;
- відвантаження готової продукції.

Складське господарство пивзаводу має бути організовано таким чином, щоб забезпечити виконання цих завдань у найкоротші терміни та з мінімальними витратами[2].

У сучасних умовах складське господарство пивзаводів все частіше автоматизується. Впровадження автоматизованих систем управління складом дозволяє підвищити ефективність роботи складу та мінімізувати помилки[3].

1.2 Аналіз готових рішень

Для автоматизації обліку та управління на підприємствах використовуються різноманітні готові ERP-системи, серед яких можна виділити Oracle, SAP, Microsoft Dynamics, 1С та інші. Розглянемо плюси та мінуси деяких з них.

SAP

Плюси

- широкий спектр функціональних можливостей, що дозволяє автоматизувати всі складські операції, від приймання товарів до відвантаження готової продукції;
- можливість інтеграції з іншими системами підприємства, наприклад, з системою управління виробництвом або системою управління продажами;
- міжнародна підтримка, що дозволяє використовувати систему на підприємствах у різних країнах світу.

Мінуси

- висока вартість впровадження та обслуговування;
- складність використання, вимагає попереднього навчання персоналу[4].

ORACLE

Плюси

- широкий спектр функціональних можливостей, що дозволяє автоматизувати всі складські операції, від приймання товарів до відвантаження готової продукції;

- можливість інтеграції з іншими системами підприємства, наприклад, з системою управління виробництвом або системою управління продажами;
- міжнародна підтримка, що дозволяє використовувати систему на підприємствах у різних країнах світу;

Мінуси

- висока вартість впровадження та обслуговування;
- складність використання, вимагає попереднього навчання персоналу[5].

MS Dynamics

Плюси

- широкий спектр функціональних можливостей, що дозволяє автоматизувати всі складські операції, від приймання товарів до відвантаження готової продукції;
- можливість інтеграції з іншими системами підприємства, наприклад, з системою управління виробництвом або системою управління продажами;
- доступна вартість впровадження та обслуговування;
- простота використання, не вимагає попереднього навчання персоналу.

Мінуси

- обмежений спектр функціональних можливостей, не дозволяє автоматизувати всі складські операції;
- неможливість інтеграції з іншими системами підприємства[6].

Системи управління складським господарством є ефективним способом автоматизації складських операцій. Вони дозволяють підвищити продуктивність, знизити ймовірність помилок та покращити контроль над складськими запасами. Але все ж ці системи не є такими гнучкими, певним підприємствам потрібні функції, які дані системи можуть надати лише частково або не надати взагалі.

1.3 Організація виробництва

1.3.1 Впровадження в експлуатацію

Повний спектр робіт відповідно до побажань замовника, починаючи від інсталяції, адаптації та налагодження програмного забезпечення і до інтеграції з пристроями і передачі в експлуатацію, називається впровадженням ПЗ в систему. Час і вартість комплексу робіт залежать від безлічі факторів і критеріїв виконання, зазначених замовником або необхідних для стабільності, таких як:

- готовність персоналу компанії до переходу на нове ПЗ або його освоєння;
- наявність необхідних для виконання апаратних засобів;
- особливостей виконання роботи;
- масштабу передбачуваних дій;
- стану баз даних на поточний момент, наявності резервних копій на крайній випадок;
- наявності і працездатності каналів зв'язку.

Оскільки процедура впровадження ПО може викликати перебої в роботі компанії, процес поділяється на кілька етапів, кожен з яких має свої нюанси і здійснюється після суворого узгодження з замовником.

Обстеження компанії

Перед створенням проекту виконується дослідження поточної роботи компанії професіоналами. Після закінчення попереднього обстеження і аудиту замовник отримує рекомендації, пов'язані з розробкою технічного завдання на проведення робіт. У ньому приділяється увага кожній деталі, детально описані вимоги по:

- підготовці та вимогам до техзасобів;
- формату зберігання і передачі даних і резервних архівів;
- складу і виконанню підготовчих робіт для об'єкта;
- конфігурації системи передачі інформації;

- роботі загального і прикладного програмного забезпечення.

Якісно складене ТЗ гарантує точність виконання робіт.

Складання контракту на виконання робіт

Контракт на проведення робіт складається за спільною згодою замовника і компанії після виконання аналізу ТЗ.

Цей період - оцінний. Оскільки план робіт призначений і терміни визначені, компанія-виконавець може оцінити всю процедуру в комплексі і визначитися з ціною. Найчастіше первинний етап проводиться безкоштовно або стає таким на підставі подальшого замовлення. Ціна на виконання робіт по інтеграції програмного забезпечення може залежати від наступних факторів:

- складу і кількості робочих місць, підсистем і модулів;
- проведення додаткових робіт по інтеграції з іншими підсистемами і системами, а також складності її виконання;
- обсягу інформації, що зберігається в БД інформації та її стану (працездатності та наявності резервних копій).

Створення групи по впровадженню ПЗ

Третій період також входить в підготовчі роботи. Компанією-виконавцем формується група впровадження програмного забезпечення і призначаються відповідальні.

Інсталяція та налагодження ПЗ

В цей період проводиться інсталяція програмного забезпечення на серверах і клієнтських машинах, підключення зв'язку, а також перевірка і налагодження робочого стану системи і її тестування під навантаженням. В стандартний перелік робіт четвертого етапу входять:

- встановлення і підготовка загальносистемного ПЗ сервера;
- інсталяція та налагодження компонентів і функцій серверної платформи;
- створення таблиць баз даних, завантаження інформації та інтеграція;

- перенесення БД (при необхідності), конвертація в потрібний формат, налагодження та створення робочих копій ПЗ, підготовка програм;
- установка і підготовка клієнтських машин (загально-прикладне і прикладне ПЗ);
- інтеграція і адаптація з уже наявними системами і платформами;
- перевірка працездатності всієї системи, тестування функціонування комплексу програмного забезпечення;
- остаточне налаштування за результатами тестування з метою отримання максимальної продуктивності і оптимізації роботи.

Завершення впровадження ПО включає виконання наступних робіт:

- навчання групи фахівців з боку замовника роботі з новим ПЗ - може проводитись віддалено або на території замовника;
- внесення змін відповідно до досвіду експлуатації замовником нового ПЗ;
- після закінчення внесення змін і усунення зауважень підписується акт здачі робіт і приймання проекту згідно ТЗ, після чого система передається замовнику і операція по впровадженню вважається завершеною.

Після інтеграції програмного забезпечення з боку замовника можуть виникнути проблеми. Це може бути людський фактор або недостатня оптимізація і інтеграція з незаявленими в ТЗ системами, які стосуються впровадженого ПО. У зв'язку з цим компанії надають технічну підтримку як своїх, так і інтегрованих сторонніми компаніями систем. Підтримка і супровід роботи серверів не входить в оплату за основні роботи, вказаними в технічному завданні.

1.3.2 Сутність, напрямки, цілі і завдання організації праці

Організація праці - це впорядкована система взаємодії працівників із засобами виробництва та один з одним, спрямована на досягнення кінцевих соціально-економічних результатів.

В умовах ринкової економіки, де конкуренція вимагає від підприємств підвищення продуктивності праці та якості продукції, значення організації праці зростає.

Основні завдання організації праці включають:

Технологічні:

- вдосконалення структури підприємства, спеціалізації виробництва, та вибір технологічних процесів;

- розробка і впровадження нової техніки і технологій;
- механізація і автоматизація трудомістких процесів;
- вдосконалення організації виробництва і праці;

Економічні:

- підвищення продуктивності праці, зниження собівартості продукції, та підвищення рентабельності виробництва;

- ефективне використання матеріальних, трудових і фінансових ресурсів;

- скорочення витрат на виробництво і збут продукції.

Психофізіологічні:

- створення сприятливих умов праці, що забезпечують високу працездатність працівників;

- збереження здоров'я працівників;
- підвищення мотивації праці;

Соціальні:

- забезпечення змістовності і привабливості праці;
- підвищення соціальної захищеності працівників;
- задоволення потреб працівників у праці, відпочинку, культурі та дозвіллі.

Для вирішення цих завдань необхідно комплексно використовувати всі напрямки організації праці, такі як:

Розробка раціональних форм поділу і кооперації праці:

- горизонтальний поділ праці (розподіл праці на окремі операції, процеси, роботи);
- вертикальний поділ праці (поділ праці на різні рівні управління);
- комбінований поділ праці (поєднання горизонтального і вертикального поділу праці).

Організація робочих місць:

- оснащення робочих місць необхідними засобами виробництва, предметами праці, інструментом;
- раціональне розміщення робочих місць;
- створення безпечних і комфортних умов праці;

Обслуговування робочих місць:

- забезпечення робочих місць необхідними матеріалами, інструментами, інформацією;
- вивезення відходів виробництва.

Оптимізація трудового процесу:

- розробка і впровадження раціональних методів і прийомів праці;
- скорочення непродуктивних витрат часу.

Створення сприятливих умов праці:

- дотримання норм і правил охорони праці, техніки безпеки і виробничої санітарії;
- забезпечення працівників необхідним обладнанням, інструментом, спецодягом та іншими засобами індивідуального захисту;
- створення комфортних умов праці (освітлення, вентиляція, шумоізоляція тощо).

Атестація і раціоналізація робочих місць:

- періодичний огляд робочих місць на предмет їх відповідності сучасним вимогам;
- розробка заходів щодо вдосконалення організації праці на робочих місцях.

Нормування праці:

- встановлення норм праці (норми часу, норми виробітку, норми обслуговування тощо);
- визначення міри праці (кількість виробленої продукції, обсяг виконаних робіт тощо).

Основна проблема організації праці - це скорочення часу виробництва продукції. Для вирішення цієї проблеми важливо вдосконалювати технологічні процеси та обладнання, розробляти раціональні форми поділу і кооперації праці, організовувати робочі місця ефективно, створювати сприятливі умови праці та комплексно використовувати всі напрямки організації праці.

1.3.3 Технолого-нормувальна карта розробки інформаційної системи

Технолого-нормувальна карта (табл.1.1) розглядається як основний документ щодо поліпшення як в цілому організації праці на робочому місці, так і його нормування. При такому варіанті нормування праці значення норм часу пов'язують з технологією виконання робіт. У додатку до технолого-нормувальної карти наводиться рекомендована оснащеність робочого місця обладнанням, машинами, механізмами, технологічним оснащенням, ручними і механізованими інструментами, контрольно-вимірювальними приладами та іншими засобами, а також планування, система обслуговування, умови і безпеку праці на робочому місці. Трудовий процес, затверджений в технолого-,нормувальній карті, повинен стати еталоном при виконанні роботи, виключивши її виконання по інтуїції, досвіду. Пропонується повне охоплення даними картами всіх робочих місць.

Таблиця 1.3.1 Технолого – нормувальна карта

Найменування операції	Тривалість виконання в нормо-годинах
1) Дослідження предметної області	40
2) Аналіз вимог та визначення функціональності	90
3) Проектування системи	120

4) Розробка бази даних	110
5) Розробка ПЗ	200
6) Тестування та відлагодження	150
7) Впровадження та навчання персоналу	100
8) Підтримка та оптимізація	

Продовження таблиці 1.3.1

Всього	810
--------	-----

1.3.4 Розрахунок чисельності розробників

В плануванні діяльності підприємств важлива роль відводиться визначенню потреб у трудових ресурсах. Процес планування трудових ресурсів на підприємствах розпочинається з аналізу наявності робочих місць для працівників відповідної кваліфікації та рівня їх використання. Кількість персоналу в організації залежить від різноманітних чинників, таких як характер, масштаб, складність та трудомісткість виробничих чи управлінських процесів, а також ступінь їхньої механізації, автоматизації та комп'ютеризації. Ці аспекти визначають нормативну (планову) чисельність. Об'єктивніше персонал може бути оцінений за обліковою (фактичною) чисельністю, тобто за кількістю працівників, які офіційно працюють в організації на даний момент. Порядок обліку персоналу регулюється Інструкцією зі статистики персоналу і заробітної плати робітників та службовців на підприємствах, в установах і організаціях. Цей документ визначає методи визначення облікового складу, середньооблікової чисельності працівників, перелік категорій персоналу, а також порядок обліку прийому і вибуття кадрів та інші аспекти.

Розрахунок необхідної чисельності робітників здійснюється за формулою:

$$N = \frac{T}{F_{\text{ef.}} \cdot K \cdot B \cdot H} \quad (1.3.1)$$

Де T - трудомісткість виготовлення продукції або виконання робіт в нормогодинах (в нормо годинах)

$F_{\text{эф.}}$ – ефективний фонд робочого часу в плановому періоді (год.)

$K.B.H$ – коефіцієнт виконання норм, від 1.11 – 1.15

$$F_{\text{эф.}} = D * \varphi * \eta \quad (1.3.2)$$

Де φ – тривалість робочої зміни (в годинах)

η – коефіцієнт, що враховує втрати робочого часу, варіюєця 0.93 – 0.96

D – кількість робочих днів

D - 35 днів

T – 810 нормо-годин

$\varphi = 8$ годин

$N = ?$

$$F_{\text{эф.}} = 35 \times 8 \times 0.96 = 268,8 \text{ год}$$

$$N = \frac{810}{268,8 \times 1,11} = 2,739 = 3 \text{чол.}$$

Для розробки програмного забезпечення нам необхідно три працівники за таким фахом:

- 1) Інженер – програміст
- 2) Тестувальник програмного забезпечення
- 3) Аналітик

Інженер – програміст – фахівець, що застосовує принципи програмної інженерії до проектування, розробки, тестування та оцінки програмного забезпечення та комп'ютерних систем.

Тестувальник програмного забезпечення - фахівець, який займається перевіркою працездатності, якості, безпеки та доступністю використання програмного забезпечення, сайтів, додатків тощо.

Аналітик – людина, яка відповідає за аналіз бізнес-вимог, розробляє на їх основі технічне завдання, функціонал/архітектуру/інтерфейс майбутньої ІС. Визначає структуру бази даних, при необхідності уточнює вимоги разом з замовником та встановлює комунікацію між ним та членами команди.

1.4 Аналіз невирішених питань та постановка завдань досліджень

Складське господарство пивзаводу є важливою частиною його виробничого процесу. Воно забезпечує зберігання сировини, матеріалів, готової продукції та інших матеріальних цінностей.

Ефективність роботи складського господарства впливає на загальну результативність діяльності пивзаводу. Тому важливо постійно аналізувати стан складського господарства та визначати напрямки його вдосконалення.

Аналіз невирішених питань

У складському господарстві пивзаводу існує ряд невирішених питань, які можуть негативно впливати на його ефективність. До таких питань можна віднести:

- недостатня автоматизація складських операцій. Багато складських операцій, таких як приймання, зберігання, відвантаження, облік продукції, виконуються вручну. Це призводить до зниження продуктивності праці, збільшення витрат часу та помилок[7];
- неоптимальна організація складського простору. Неправильна розстановка обладнання та матеріалів на складі може призвести до ускладнення роботи складських працівників, зниження пропускнуєї здатності складу та погіршення умов зберігання продукції[8];
- недостатній контроль над запасами. Неякісний облік запасів може призвести до їх нестачі або надлишку, а також до втрат матеріальних цінностей[9].

Постановка завдань досліджень

Для вирішення невирішених питань складського господарства пивзаводу необхідно провести дослідження, спрямовані на:

- розробку та впровадження автоматизованої системи управління складським господарством. Автоматизована система управління складським господарством дозволить автоматизувати складські операції, підвищити продуктивність праці та зменшити витрати часу;

- оптимізацію організації складського простору. Оптимізація організації складського простору дозволить підвищити пропускну здатність складу, покращити умови зберігання продукції та зменшити витрати на складські операції;
- удосконалення системи обліку запасів. Удосконалення системи обліку запасів дозволить забезпечити своєчасне виявлення нестач та надлишків запасів, а також попередити втрати матеріальних цінностей;

Конкретні завдання досліджень

Кожен з цих напрямів досліджень передбачає постановку конкретних завдань. Наприклад, для розробки та впровадження автоматизованої системи управління складським господарством необхідно вирішити такі завдання:

- визначення складу та функціональних можливостей системи;
- розробка алгоритмів та програм для виконання складських операцій;
- впровадження системи в експлуатацію[7].

Для оптимізації організації складського простору необхідно вирішити такі завдання:

- аналіз існуючої організації складського простору;
- розробка проектів оптимізації організації складського простору;
- впровадження проектів у експлуатацію[2].

Для удосконалення системи обліку запасів необхідно вирішити такі завдання:

- аналіз існуючої системи обліку запасів;
- розробка проектів удосконалення системи обліку запасів;
- впровадження проектів у експлуатацію[9].

Результати досліджень у цих напрямках дозволять підвищити ефективність роботи складського господарства пивзаводу та забезпечити його відповідність сучасним вимогам.

Висновок

В даному розділі було проаналізовано сучасний стан інформаційної підтримки складського господарства пивзаводу та системи які можна потенційно ввести, охарактеризовано що таке складське господарство та як організоване виробництво. Було визначено сутність, напрямки цілі і завдання організації праці та створено технолого-нормувальну карту. Розраховано потрібну чисельність робітників для розробки ІС та поставлено завдання досліджень.

2. ПРОЕКТУВАННЯ ІС СКЛАДСЬКОГО ГОСПОДАРСТВА ПИВЗАВОДУ

2.1 Теоретичні передумови моделювання інформаційної системи

В основу розробки інформаційної системи складського господарства пивзаводу покладено задачу комівояжера. Задача комівояжера (TSP, Travelling Salesman Problem) - це класична задача оптимізації, яка полягає в знаходженні найкоротшого маршруту, що проходить через кожну точку заданої множини точок лише один раз.



Рисунок 2.1. Приклад знаходження найкоротшого шляху через 15 міст
Німеччини

У контексті складського господарства пивзаводу, задача комівояжера може бути використана для розробки маршрутів для доставки сировини від постачальника до складу, а також для розробки маршрутів для доставки готової продукції зі складу до клієнтів.

Розглянемо це більш детально на прикладах

Нехай у пивзаводу є постачальник сировини, який знаходиться у певному місці, а на складі є спеціальні склади для зберігання сировини. Завданням є розробка такого маршруту доставки сировини, який буде найкоротшим, тобто відстань від постачальника до заводу повинна бути мінімальною.

Цю задачу можна представити як задачу комівояжера, де кожній точці відповідає склад або постачальник, а відстань між точками відповідає відстані між ними.

Для вирішення цієї задачі можна використовувати алгоритм жадібного вибору. Алгоритм працює наступним чином:

Починаємо з будь-якої точки.

Потім, вибираємо наступну точку таким чином, щоб відстань між поточним складом та наступним складом була мінімальною.

Повторюємо кроки 2-3, поки не відвідаємо всі точки.

У нашому випадку, алгоритм жадібного вибору почне з будь-якої точки (постачальника або складу) та буде відвідувати наступні точки таким чином, щоб відстань між ними була мінімальною[10].

За таким самим алгоритмом можна представити завдання з доставки готової продукції з складів пивзаводу до клієнтів. Суть завдання та його вирішення будуть таким як і при постачанні сировини.

Задача комівояжера є потужним інструментом, який може бути використаний для розробки маршрутів для доставки сировини та готової продукції у складському господарстві пивзаводу. Використання задачі комівояжера дозволяє мінімізувати загальну відстань, яку потрібно буде подолати, що може призвести до зниження витрат на логістику.

2.2 Моделювання та конструювання

2.2.1 Розробка моделі системи

Аналіз вимог базується на застосуванні варіантів використання системи (Use Cases), які охоплюють як функціональні, так і нефункціональні вимоги, специфічні для конкретних сценаріїв використання. Для забезпечення задоволення користувача важливим є виконання системою конкретних дій, під час яких він взаємодіє з нею згідно своїх потреб. Таким чином, аналізуючи всі можливі варіанти використання системи різними користувачами або іншими зовнішніми процесами, ми визначаємо функціональні вимоги до неї.

Враховуючи, що кожен конкретний користувач взаємодіє з системою унікальним чином, для ідентифікації реальних варіантів використання системи необхідно глибоко розуміти потреби всіх зацікавлених користувачів, аналізувати накопичену інформацію і систематизувати її в актуальні сценарії використання системи, відділяючись від конкретних користувачів та враховуючи бізнес-завдання.

На підставі вимог визначається структура та форма користувацьницького інтерфейсу для реалізації конкретного прецеденту. Для уточнення вимог до програмного продукту ми будемо Use-Case діаграму в уніфікованій мові моделювання Rational Rose.

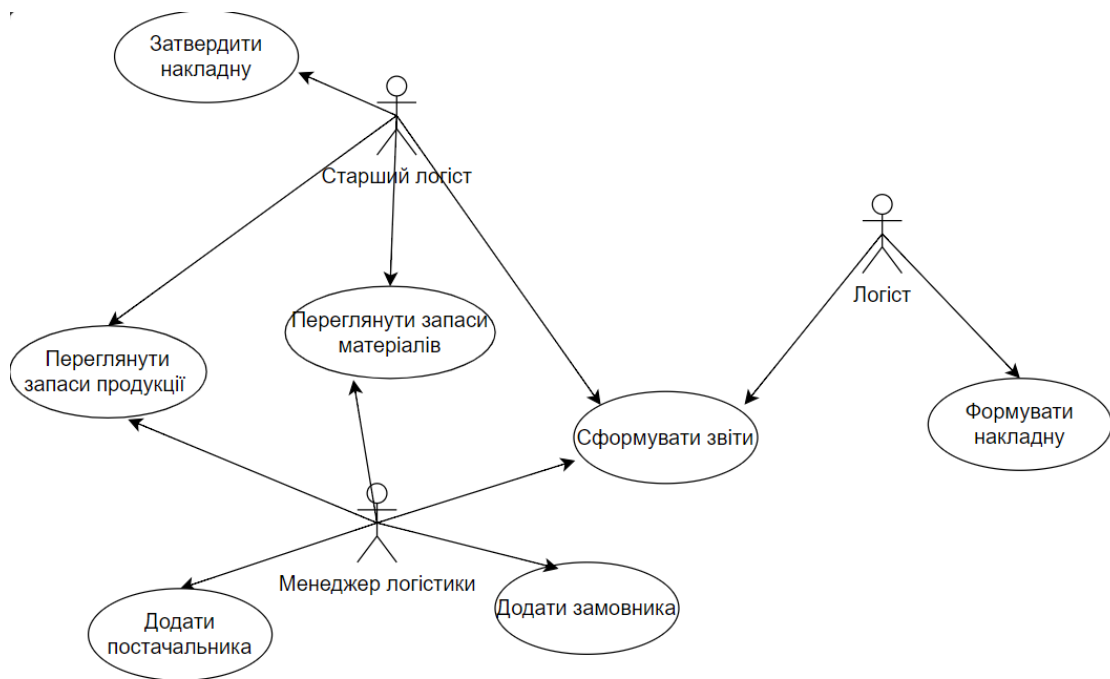


Рисунок 2.2 Use-case діаграма [складено автором]

Користувачами (акторами на схемі) є менеджер логістики, старший логіст, та звичайний працівник логістичної служби (логіст). Кожен з них виконує певні функції та має певні рівні доступу до системи. Так наприклад менеджер має доступ до всієї системи та може робити всі доступні дії всередині системи.

Старший логіст має в свою чергу такі варіанти використання:

- перегляд запасів матеріалів;
- перегляд запасів продукції;
- затвердження накладних;

- формування звітів;
- формування накладних.

Працівник-логіст має найменше прав та відповідно найменше варіантів використання:

- формування звітів;
- формування накладних.

Use case діаграма дозволяє узгодити із замовником функції майбутньої системи. Переглядаючи діаграму, Замовник може надати пропозиції по включенню додаткових функцій або розширенню представлених. В результаті уточнень на наступних етапах розробки програмного продукту ймовірність виникнення змін у вимогах зменшиться, що зменшує ризики розробки проекту. Use-case діаграма є основою для розробки технічного завдання.

2.2.2 Дослідження потоків даних

Ми проведемо дослідження потоків даних, використовуючи діаграми потоків даних. Ці діаграми є графічними моделями, які відображають "потoki" даних в інформаційній системі. Вони служать для візуалізації обробки даних (структурного проектування) в системі. Подальше уточнення діаграм відбуватиметься шляхом деталізації процесів і потоків даних з метою надання більш детального зображення розроблюваної системи.

На рисунку 2.3 приведена діаграма нульового рівня. Сутності «Менеджер логістики», «Старший логіст», «логіст» взаємодіють з процесом IC «BrewSystem»

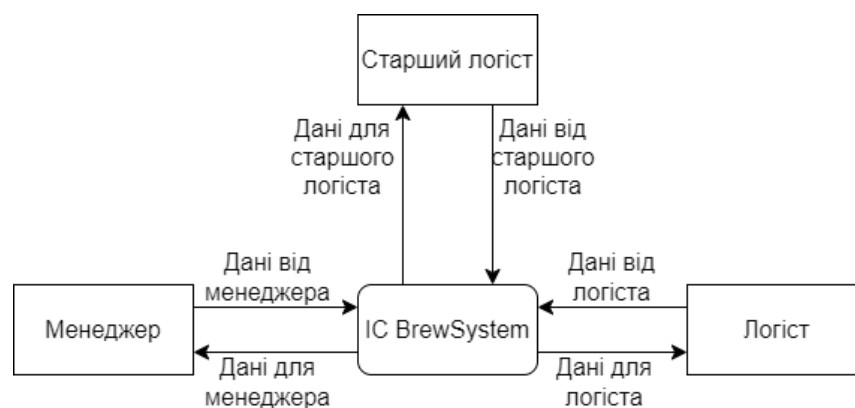


Рисунок 2.3. ДПД 0-го рівня [створено автором]

Зміст інформаційних потоків ДПД нульового рівня наступний:

Дані для менеджера:

- Дані про переміщення
- Виробничі потужності

Дані від менеджера

- Дані про замовників
- Дані про постачальника
- Вартість продукції
- Коригування асортименту
- затверджені накладні

Дані для старшого логіста:

- Дані про запаси
- Дані про преміщення

Дані від старшого логіста:

- Коригування асортименту
- виробничі потужності
- затверджені накладні

Дані для логіста:

- Перелік замовлень
- Обсяги поставок
- Дані про постачальників
- Дані про замовників

Дані від логіста:

- Інформація про завантаження
- Інформація про відвантаження

Побудуємо діаграму першого рівня, яка дає уявлення про функціонування системи. Процес першого рівня містить в собі наступні процеси: «Формувати/коригувати довідники», «Обробити дані відвантаження», «Обробити дані завантаження».

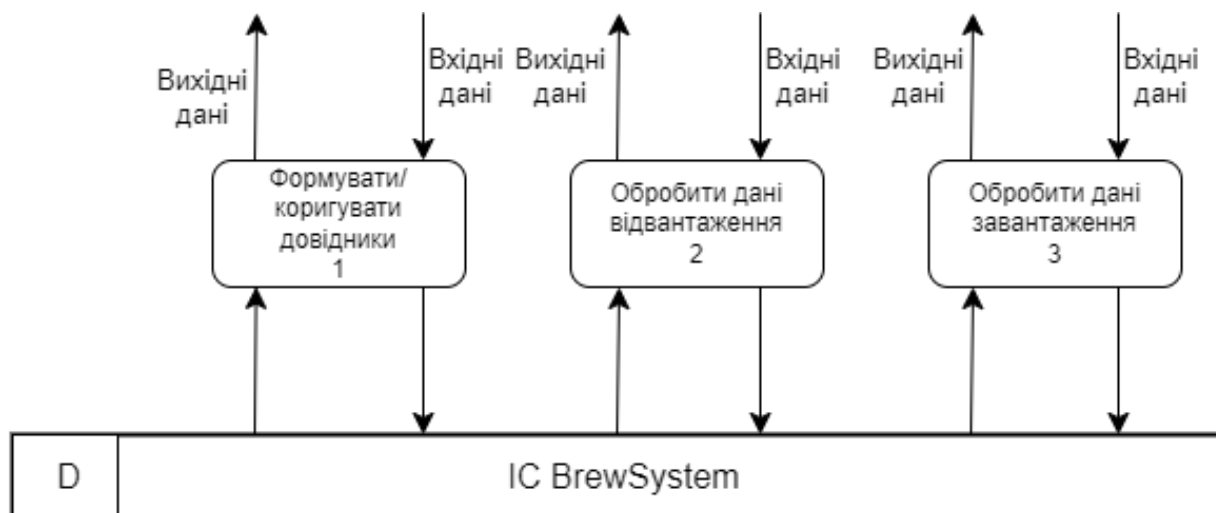


Рисунок 2.4. ДПД 1-го рівня [створено автором]

Таблиця 2.1 Словник даних ДПД 1-го рівня [створено автором]

ДПД 0-го рівня	ДПД 1-го рівня
Інформація від менеджера	Вхідна інформація
Інформація від старшого логіста	
Інформація від логіста	
Інформація для менеджера	Вихідна інформація
Інформація для старшого логіста	
Інформація для логіста	

Процес «Обробки даних завантаження» представлено на рисунку 2.5 дає основне уявлення про функціонування процесу. Він складається з під процесів: «Отримання замовлення від клієнта», «Перевірка наявності товару на складі», «Формування накладної відвантаження».

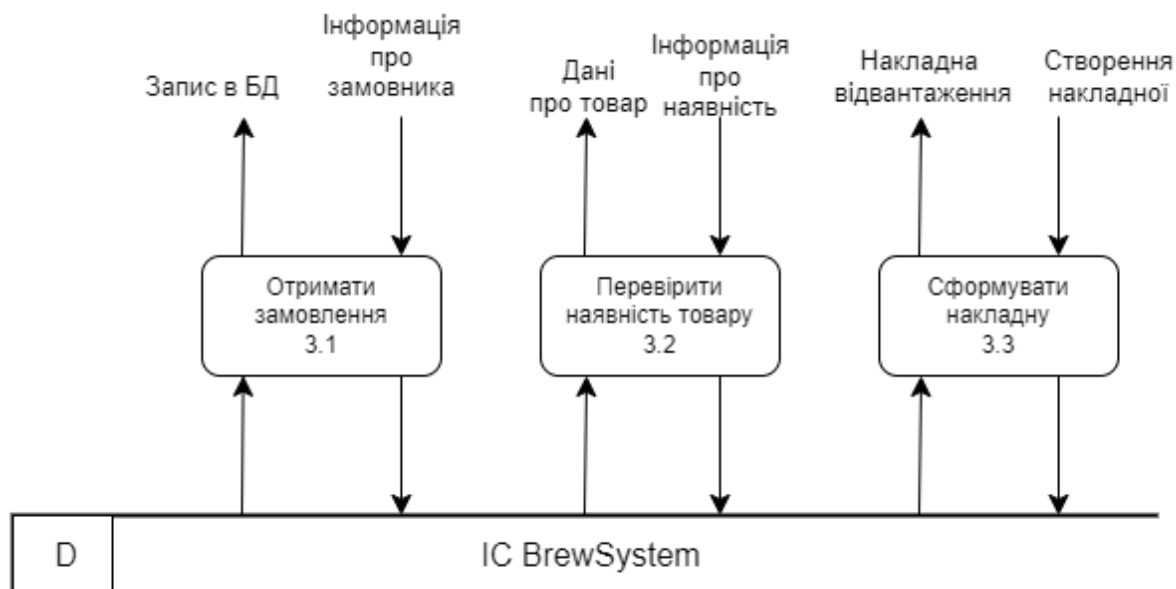


Рисунок 2.5. ДПД 2-го рівня [створено автором]

Таблиця 2.2 Словник даних ДПД 2-го рівня [створено автором]

ДПД 1-го рівня	ДПД 2-го рівня	Атрибути
1	2	3
Вхідні дані	Інформація по замовника	Ім'я Телефон Ел. Адреса Індекс Адреса
	Інформація про наявність продукту	Тип продукту Назва продукту
	Дані для створення накладної	Дата створення Дата відвантаження Назва продукту Замовник Кількість

Продовження таблиці 2.2

Вихідні дані	Запис в БД про замовника	Ім'я Телефон Ел. Адреса Індекс Адреса
	Дані про товар	ІД продукту Тип продукту Назва продукту Ціна продукту
	Готова накладна	Дата створення Дата відвантаження Співробітник Назва продукту Замовник Кількість Адреса Індекс Телефон

Процес створення накладної представлено на рисунку 2.6 дає основне уявлення про відвантаження готової продукції. Цей процес складається з процесу «Аналізу даних наявності продукції» та «Формування накладної відвантаження».

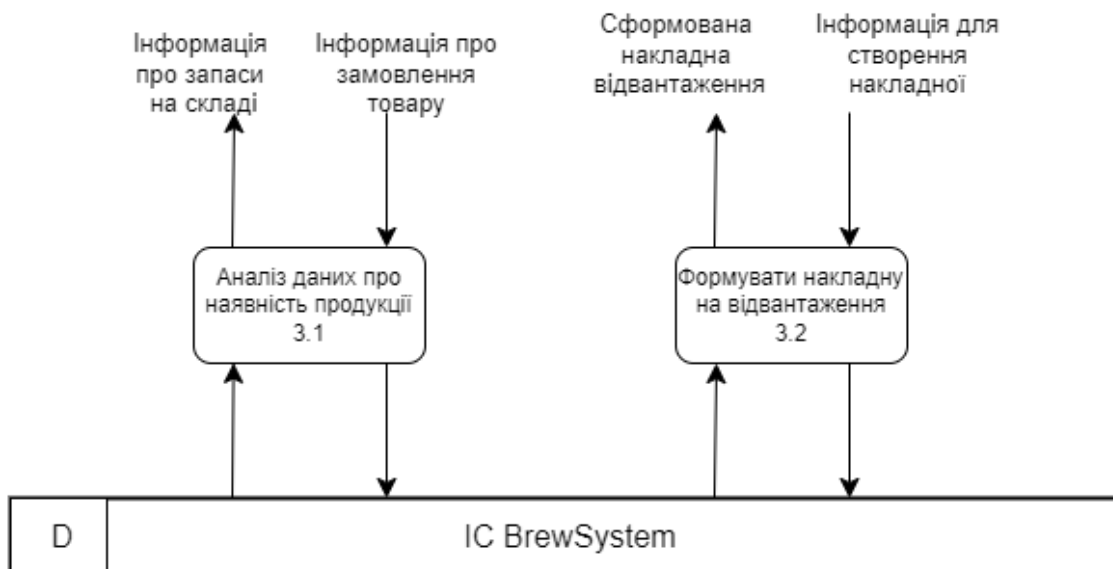


Рисунок 2.6. ДПД 3-го рівня [створено автором]

Таблиця 2.3 Словник даних ДПД 3-го рівня [створено автором]

ДПД 2-го рівня	ДПД 3-го рівня	Атрибути
1	2	3
Вхідні дані	Інформація про замовлення	Тип товару Назва товару Кількість товару
	Інформація для створення накладної	Співробітник Назва продукту Замовник Кількість Адреса Індекс Телефон
Вихідні дані	Інформація про запаси	Тип товару Назва товару Кількість доступного товару

Продовження таблиці 2.3

	Сформована накладна відвантаження	Дата створення Дата відвантаження Співробітник Назва продукту Замовник Кількість Адреса Індекс Телефон Р/Р банку МФО ЄДРПОУ
--	---	--

2.2.3 Концептуальна модель

Побудова концептуальної моделі є процесом створення інформаційної моделі предметної області з врахуванням поставлених вимог до цього процесу. Результати аналізу інформаційних потоків предметної області дають можливість сформулювати концептуальну модель даних, що є першим кроком у розробці будь-якої бази даних. На етапі побудови концептуальної моделі використовуються попередньо визначені сутності та їх атрибути. В цій моделі відображаються зв'язки між сутностями, визначаються ключові атрибути та типи зв'язків. Сутність розглядається як інформаційний об'єкт, який зберігається в базі даних, а атрибут представляє властивість цієї сутності. Зв'язки сутностей визначають залежність між атрибутами цих сутностей, можуть існувати між двома чи більше сутностями.

Після дослідження предметної області було виявлено такі сутності:

- 1) Накладна
- 2) Постачальники ресурсів
- 3) Ресурси

- 4) Замовники
- 5) Готова продукція
- 6) Співробітники
- 7) Маршрутний лист

Сутності пов'язані між собою зв'язками один – до – багатьох, багато – до – одного та один – до – одного (рис 2.7).

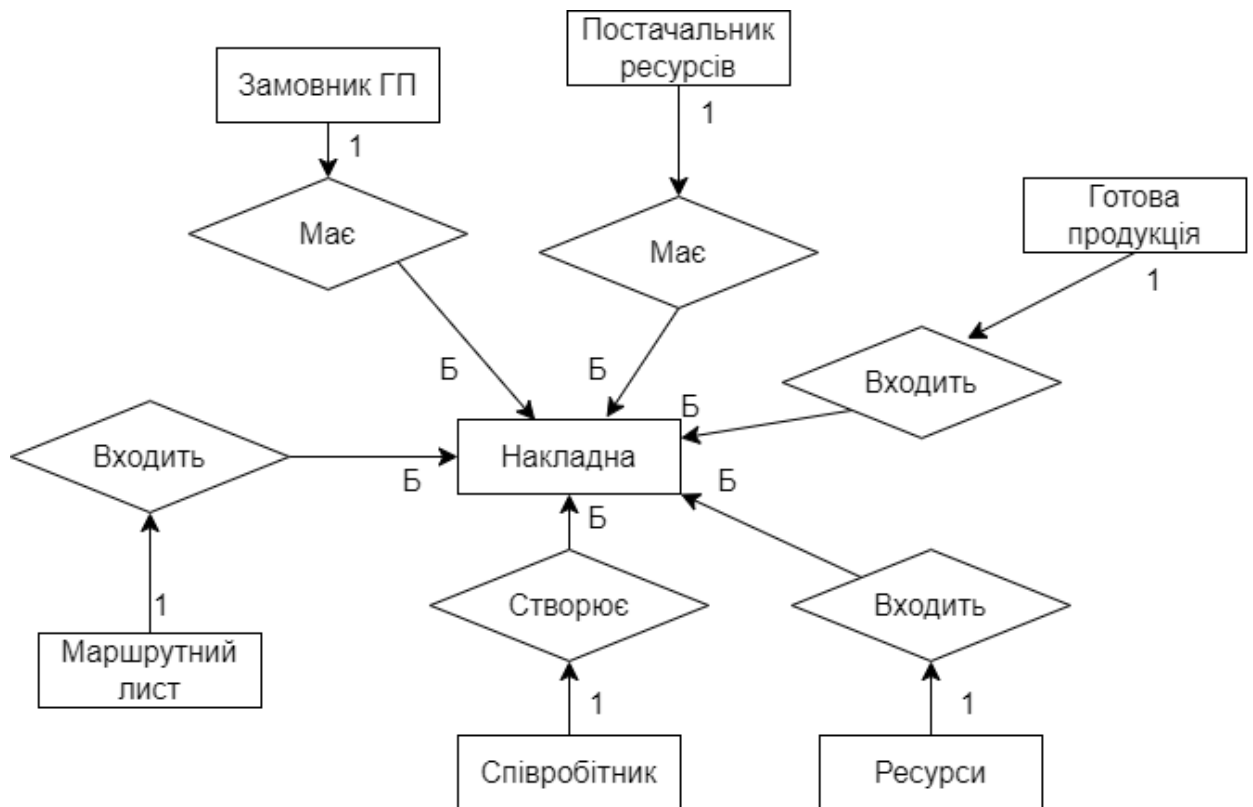


Рисунок 2.7. Концептуальна модель даних [створено автором]

2.2.4 Розробка діаграми станів програми

Діаграма станів визначає можливі послідовності станів і переходів, які узагальнено відображають поведінку елемента моделі протягом його життєвого циклу. Ця діаграма описує динамічну поведінку сутностей, базуючись на специфікації їх реакції на конкретні події.

Хоча діаграми станів часто використовуються для опису поведінки окремих екземплярів класів (об'єктів), вони також можуть служити для специфікації функціональності інших компонентів моделей, таких як варіанти використання, актори, підсистеми, операції і методи.

Діаграма станів в суті є графом спеціального виду, що представляє автомат. У контексті UML термін "автомат" має конкретну семантику, ґрунтуючись на теорії автоматів. Вершини графа представляють стани і інші елементи автомата (псевдостани), які зображаються відповідними графічними символами. Дуги графа служать для позначення переходів між станами. Діаграми станів можуть вкладатися одна в одну, утворюючи вкладені діаграми для докладнішого представлення окремих елементів моделі.

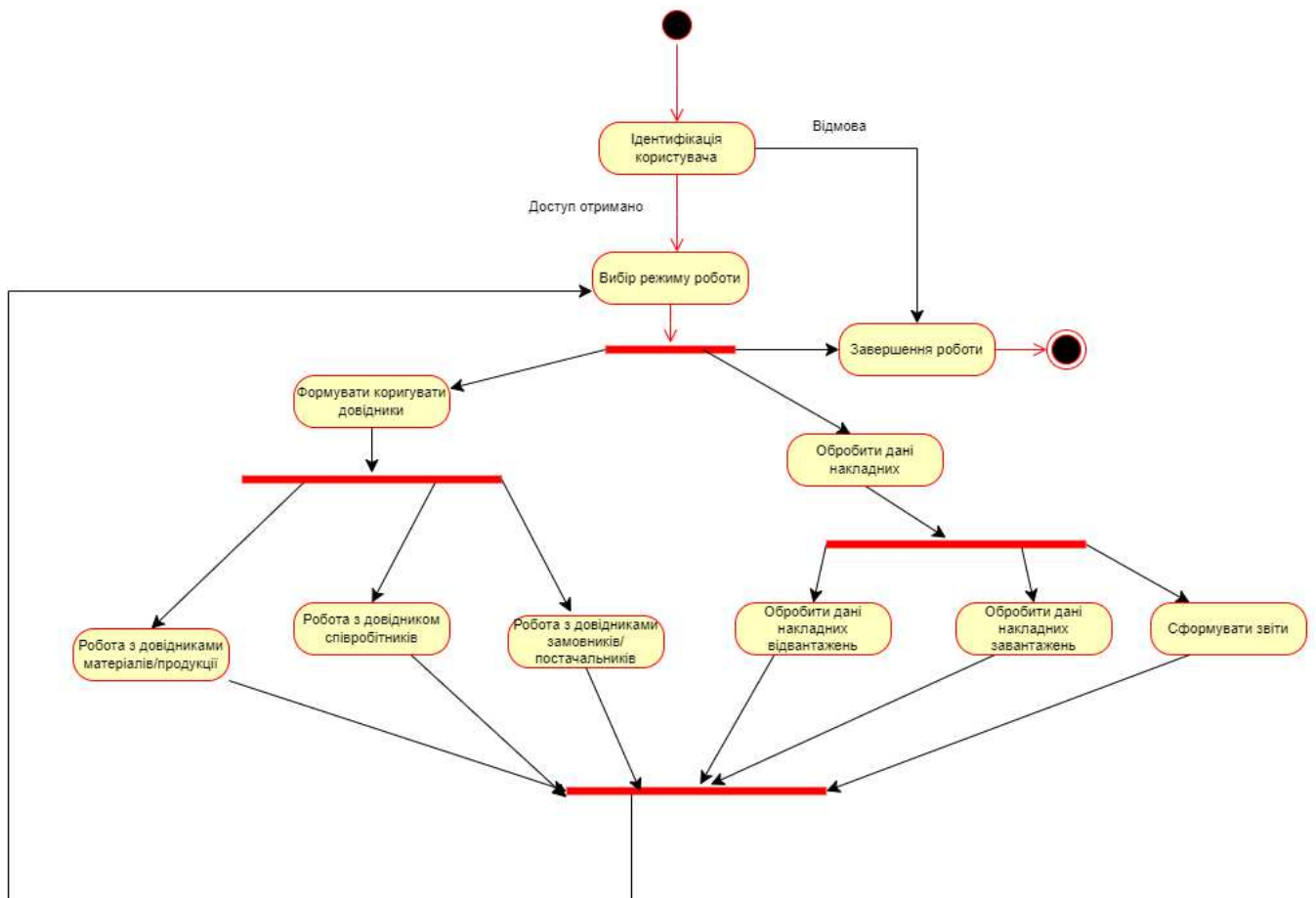


Рисунок 2.8. Діаграма станів [створено автором]

2.3 Технічний проект

2.3.1 Логічна модель даних

Процес формування інформаційно-логічної моделі вимагає глибокого розуміння предметної області та її логічних взаємозв'язків. Тому виявлення

інформаційних об'єктів стає більш ефективним, якщо цей етап проводиться спільно з користувачем та розробником бази даних інформаційної системи.

Дані про об'єкти предметної області містяться в конкретних документах, які виступають основними носіями інформації. Ці документи можуть бути представлені у друкованій або рукописній формі на бланку з визначеною формою та правовою чинністю.

Логічна модель базується на концептуальній моделі та є більш детальним її відображенням. У цьому варіанті одна сутність може мати відображення в більш ніж одній таблиці, а атрибути сутностей перетворюються в поля таблиць. Взаємозв'язки між таблицями стають більш точними, і виділяються первинні ключі.

Логічна модель даних наведена в таблицях 2.4-2.10

Таблиця 2.4. Накладна [створено автором]

Поле	Тип даних	Ключ
ІД накладної	Числовий	+
ІД співробітника	Числовий	
ІД Маршруту	Числовий	
ІД замовника	Числовий	
ІД продукту	Числовий	
ІД постачальника	Числовий	
ІД матеріалу	Числовий	
Одиниці виміру	Текстовий	
Вага	Текстовий	
Дата створення накладної	Дата	
Дата транспортування	Дата	

Таблиця 2.5. Замовники [створено автором]

Поле	Тип даних	Ключ
ІД Замовника	Числовий	+
ПІБ замовника	Текстовий	

Продовження таблиці 2.5

ПІБ директора	Текстовий	
Адреса	Текстовий	
Індекс	Текстовий	
Телефон	Текстовий	
Ел. пошта	Текстовий	
Розрах. разунок	Текстовий	
МФО	Текстовий	
ЄДРПОУ	Текстовий	

Таблиця 2.6. Постачальники [створено автором]

Поле	Тип даних	Ключ
ІД постачальника	Числовий	+
ПІБ постачальника	Текстовий	
ПІБ директора	Текстовий	
Адреса	Текстовий	
Індекс	Текстовий	
Телефон	Текстовий	
Ел. пошта	Текстовий	
Розрах. разунок	Текстовий	
МФО	Текстовий	
ЄДРПОУ	Текстовий	

Таблиця 2.7. Готова продукція [створено автором]

Поле	Тип даних	Ключ
ІД продукту	Числовий	+
Тип продукту	Текстовий	
Назва продукту	Текстовий	
Ціна	Числовий	

Продовження таблиці 2.7

Наявність на складі	Текстовий	
Одиниці виміру	Текстовий	

Таблиця 2.8. Матеріали [створено автором]

Поле	Тип даних	Ключ
ІД ресурсу	Числовий	+
Назва ресурсу	Текстовий	
Ціна	Числовий	
Наявність на складі	Текстовий	
Одиниці виміру	Текстовий	

Таблиця 2.9. Співробітники [створено автором]

Поле	Тип даних	Ключ
ІД співробітника	Числовий	+
Посада	Текстовий	
ПІБ	Текстовий	
Телефон	Текстовий	

Таблиця 2.10. Маршрутний лист [створено автором]

Поле	Тип даних	Ключ
ІД маршруту	Числовий	+
Назва	Текстовий	

Результатом створення бази даних є система, яка організована та структурована для зберігання інформації. Ця система використовується для зберігання різних типів інформації, таких як дані про клієнтів, продукти, накладні та ін.

База даних складається з таблиць, які містять дані. Таблиці пов'язані між собою за допомогою зв'язків. Зв'язки дозволяють організувати інформацію в логічну структуру, яка може бути легко оброблена[11].

Схема бази даних – це опис структури бази даних. Схема визначає, які таблиці містяться в базі даних, які поля є в кожній таблиці та які зв'язки є між таблицями.

Схема бази даних створюється розробником бази даних. Схема є важливим документом, оскільки вона визначає, як буде зберігатися інформація в базі даних. На рисунку 2.8 наведено схему даних бази даних в MySQL Workbench, яка була побудована базуючись на представленій логічній моделі[12].

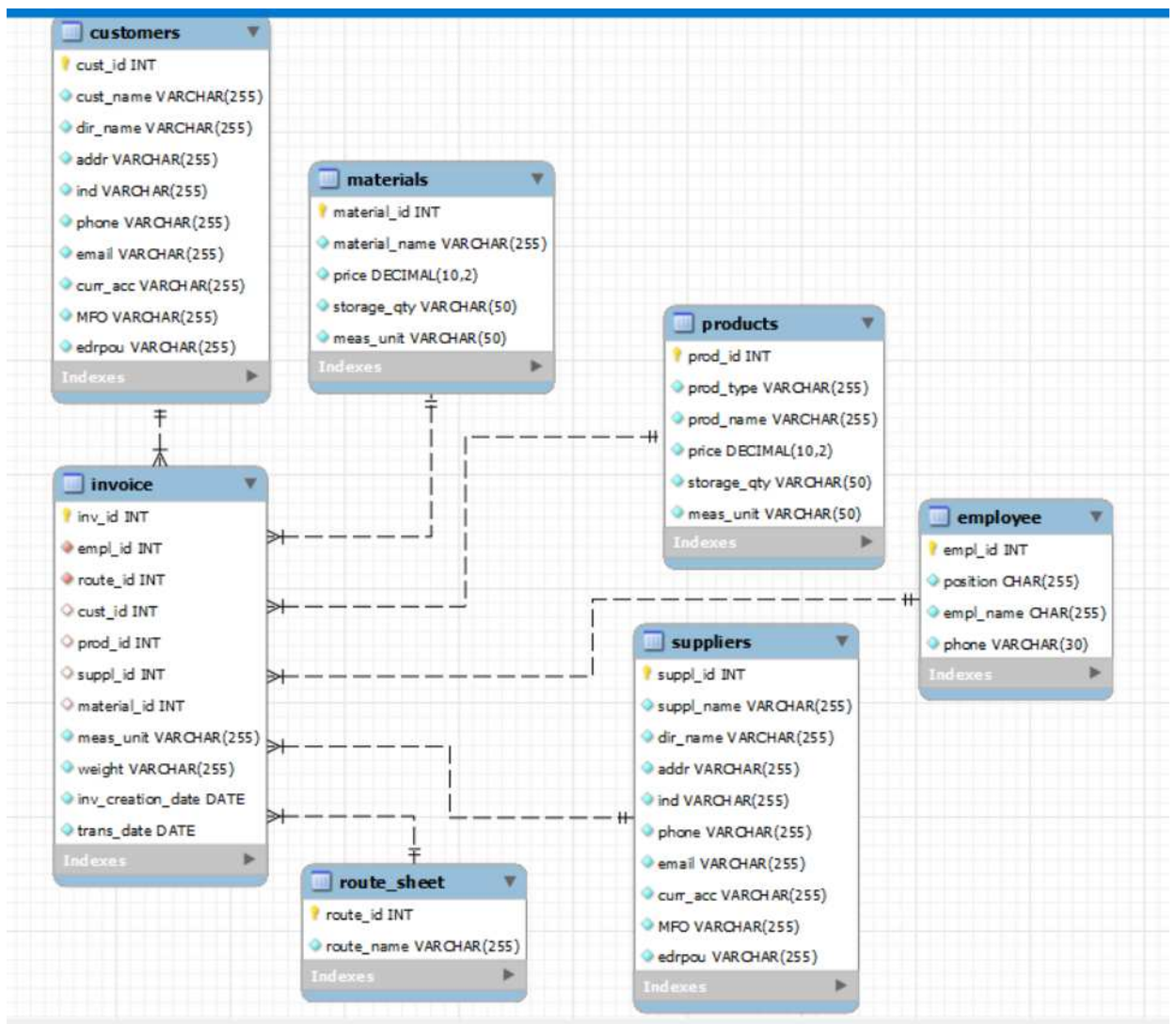


Рисунок 2.8. Фізична модель БД [створено автором]

2.3.2 Розробка архітектури програми

Архітектура програмного забезпечення складається з різноманітних структур та концепцій, що виявляють різні рівні абстракції та відображають різні аспекти (такі як структура класів програмного забезпечення, розгортання компонентів на фізичних машинах, можливі сценарії взаємодій між компонентами тощо). Ці структури взаємодіють із зв'язком, який об'єднує всі представлені дані з структурними елементами програмного забезпечення.

Архітектура програмного забезпечення представляється у вигляді окремих модулів програми, що визначаються на підставі раніше створеної діаграми використання. Кожен модуль відображає функції, які виконують актори в системі. Управління роботою програми здійснюється головним модулем, який дає користувачеві можливість вибору режиму роботи. Після вибору режиму активізується відповідна форма взаємодії.

Перелік модулів включає:

- головний модуль – перевірки прав доступу;
- модуль завершення роботи;
- модуль налаштувань;
- модуль роботи з таблицями;
- модуль роботи з запитами;
- модуль роботи з звітами.

2.3.3 Специфікація модулів

В архітектурі програми передбачено модульну структуру. Кожен модуль реалізує певну операцію.

Модуль «mainUnit», призначений для отримання доступу до програми, після якого користувач може відразу обрати необхідний режим роботи з програмою.

Вхідні дані: логін та пароль користувача.

Дія: перевірка введеного логіну та паролю.

Вихідні дані: головна форма програми якщо дані були введені коректно або повідомлення про помилку при вводі даних авторизації.

Модуль “Накладна” (invoiceUnit) викликається головним модулем.

Призначення: створення нової накладної та додавання її в таблицю.

Вхідні дані: вибір позиції на додавання нової накладної на відвантаження або завантаження

Дія: додавання/коригування запису.

Вихідна інформація: виконання транзакції в таблиці «Накладна».

Модуль “Замовники” (customersUnit) викликається головним модулем.

Вхідні дані: перегляд/редагування/доповнення інформації про замовників готової продукції

Дія: додавання/видалення/коригування запису.

Вихідна інформація: виконання транзакції в таблиці «Замовники».

Даний модуль дозволяє нам переглянути/редагувати інформацію про замовників ГП.

Модуль “Продукція” (prodUnit) викликається головним модулем.

Вхідні дані: перегляд/редагування/доповнення інформації про готову продукцію.

Дія: додавання/видалення/коригування запису.

Вихідна інформація: виконання транзакції в таблиці «Готова продукція».

Даний модуль дозволяє нам переглянути/редагувати інформацію про готову продукцію.

Модуль “Постачальники” (suppliersUnit) викликається головним модулем.

Вхідні дані: перегляд/редагування/доповнення постачальників матеріалів.

Дія: додавання/видалення/коригування запису.

Вихідна інформація: виконання транзакції в таблиці «Постачальники».

Даний модуль дозволяє нам переглянути/редагувати/додати інформацію про постачальників матеріалів.

Модуль “Матеріали” (materialsUnit) викликається головним модулем.

Вхідні дані: перегляд/редагування/доповнення матеріалів для виробництва.

Дія: додавання/видалення/коригування запису.

Вихідна інформація: виконання транзакції в таблиці «Матеріали».

За допомогою даного модуля можна переглянути/редагувати/додати інформацію про кількість матеріалів на виробництві, які наявні на складі.

Модуль “Співробітники” (emplUnit) викликається головним модулем.

Вхідні дані: перегляд/редагування/доповнення інформації про співробітників логістичної служби.

Дія: додавання/видалення/коригування запису.

Вихідна інформація: виконання транзакції в таблиці «Співробітники».

Даний модуль дозволяє нам переглянути/редагувати інформацію про співробітників логістичної служби.

Модуль “Маршрутний лист” (routeUnit) викликається головним модулем.

Вхідні дані: перегляд інформації про наявні переміщення продукції.

Дія: перегляд запису.

Вихідна інформація: виконання транзакції в таблиці «Маршрутний лист».

Даний модуль дозволяє нам переглянути інформацію наявні переміщення.

Модуль “Звіти” (printRepForm) викликається модулем «Накладна».

Вхідні дані: інформація таблиці Накладна.

Дія: створення та друк накладної.

Вихідна інформація: створена накладна.

Даний модуль дозволяє нам створити та переглянути звіт по накладній та роздрукувати її. З його допомогою можна екпортувати накладну в різні типи файлів, наприклад pdf або docx.

Модуль «settingsUnit» дозволяє користувачу налаштувати програмне забезпечення.

Вхідні дані: відсутні.

Дія: зміна інтерфейсу, мови програми або налаштування певних функцій програми.

Вихідні дані: відсутні.

2.3.4 Алгоритм роботи програми

Алгоритм — це послідовність точних команд виконавцеві алгоритму спрямованих на розв'язання певної задачі. Алгоритм поданий графічним способом у вигляді блок-схеми.

Блок-схема алгоритму — це графічне зображення алгоритму у вигляді спеціальних блоків із необхідними словесними поясненнями. Кожний етап алгоритму на блок-схемі подається у вигляді геометричної фігури, яка має певну форму залежно від характеру дії.

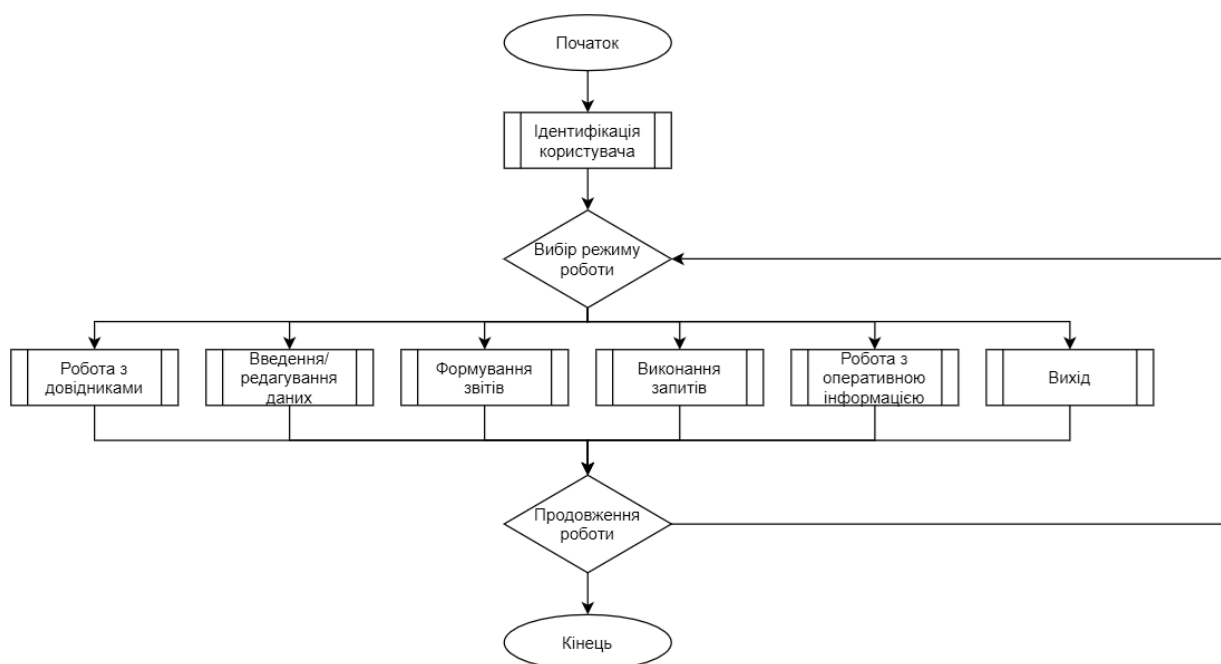


Рисунок 2.9. Алгоритм роботи програми [створено автором]

2.4 Робочий проект

2.4.1 Вибір засобів розробки програмного забезпечення ІС

У ролі платформи для розробки було обрано Delphi 11 від Embarcadero RAD Studio, яке є потужним середовищем для швидкого створення застосунків з графічним інтерфейсом для платформи Windows. Delphi 11 включає близько 200 різноманітних компонентів і, крім того, дозволяє створювати, встановлювати та використовувати власні компоненти.

Замість окремого інструмента для взаємодії з візуальними компонентами, Delphi 11 має інтегровану "Палітру компонентів". Ця палітра розділена на кілька

функціональних груп і містить бібліотеку з понад 100 візуальних компонентів, які легко перетягуються на форму і стають елементами управління для прототипування програми.

Після розміщення компоненту на формі можна редагувати їх властивості та встановлювати обробку подій об'єкту через код. Проект розробляється поетапно, зі змінами у властивостях, подіях та функціях використовуваних елементів.

IDE Delphi 11 забезпечує високу швидкодію при компіляції та складанні застосунків для платформи Windows. Час, витрачений на розробку програми, може бути ефективно відображений. В результаті отримуємо добре оптимізовані програми щодо швидкості виконання і використання пам'яті.

Borland Delphi 11 підтримує зв'язок з різними базами даних, включаючи dBASE, Paradox, Sybase, Oracle, InterBase, Informix, Excel, Access, FoxPro, MS Sql, MySQL та Btrieve. BDE (Borland Database Engine) спрощує обслуговування зв'язків з базами даних, а Database Explorer графічно відображає їхню структуру.

Delphi 11 дозволяє ефективно проводити налагодження та тестування програм завдяки інструментам керування проектами, двосторонній інтеграції візуального та текстового редагування, а також вбудованому налагоджувачу з підтримкою асемблерного вікна прокрутки, покроковим виконанням, точками зупинки та трасуванням.

Для роботи з базами даних було обрано MySQL. MySQL — це відкрита система управління базами даних (СУБД) з відкритим кодом, яка використовується для зберігання і обробки інформації. MySQL є однією з найпопулярніших СУБД у світі, і вона широко використовується для створення веб-додатків, веб-сайтів, програмних систем та інших застосунків.

MySQL має ряд переваг, які роблять її привабливою для використання в якості СУБД для створення інформаційної системи пивзаводу. Ці переваги включають:

- відкрита ліцензія: MySQL є вільною для використання, поширення та модифікації. Це робить її доступною для широкого кола користувачів, включаючи малий і середній бізнес;

- простота використання: MySQL має простий і інтуїтивно зрозумілий інтерфейс. Це робить її легкою для вивчення та використання, навіть для користувачів без досвіду роботи з СУБД;
- висока продуктивність: MySQL є високопродуктивною СУБД. Вона може обробляти великі обсяги даних і підтримувати високі рівні продуктивності;
- безпека: MySQL пропонує широкий спектр функцій безпеки для захисту даних. Це робить її безпечною для використання в критичних до безпеки додатках;

Ось деякі конкретні приклади того, як MySQL може бути використана для створення інформаційної системи пивзаводу:

- Для зберігання інформації про складські запаси: MySQL може бути використана для зберігання інформації про товари, які є на складі пивзаводу. Ця інформація може включати назву товару, кількість, термін придатності та місце зберігання;
- Для управління замовленнями: MySQL може бути використана для управління замовленнями клієнтів. Ця інформація може включати назву товару, кількість, дату замовлення та дату доставки;
- Для контролю якості: MySQL може бути використана для контролю якості продукції. Ця інформація може включати результати лабораторних тестів, дані про термін придатності та інформацію про відкликання продукції;

MySQL є хорошим вибором для створення інформаційної системи пивзаводу. Вона пропонує ряд переваг, які роблять її привабливою для використання в цьому контексті.

2.4.2 Фізична модель бази даних

Розробка фізичної моделі бази даних вбудована в принципи організації даних, які були визначені у логічній моделі бази даних.

Фізична модель визначає, як дані будуть зберігатися у зовнішній пам'яті і представляє собою внутрішню структуру системи. Ця модель також відома як

внутрішня, і її форма залежить від обраної системи керування базами даних (СКБД). У контексті використання MySQL, ідентифікатори (імена) таблиць і полів повинні відповідати вимогам цієї СКБД. Типи даних, розміри полів та обмеження також повинні бути налаштовані відповідно до стандартів MySQL.

Таблиця 2.11. Накладна [створено автором]

Логічна назва поля	Фізична назва поля	Тип	Довжина поля
Код накладної	Inv_id	INT	Ціле
Код співробітника	Empl_id	INT	Ціле
Тип маршруту	Route_id	INT	Ціле
Код замовника	Cust_id	INT	Ціле
Код продукту	Prod_id	INT	Ціле
Код постчальника	Suppl_id	INT	Ціле
Код матеріалу	Material_id	INT	Ціле
Одиниці виміру	Meas_unit	VARCHAR	20
Вага	Weight	VARCHAR	20
Дата створення	Inv_creation_date	date	Дата
Дата транспортування	Trans_date	date	дата

Таблиця 2.12. Замовники [створено автором]

Логічна назва поля	Фізична назва поля	Тип	Довжина поля
Код замовника	cust_id	INT	Ціле
ПІБ замовника	Cust_name	VARCHAR	100
ПІБ директора	Dir_name	VARCHAR	100
Адреса	Addr	VARCHAR	100
Індекс	Ind	VARCHAR	15
Телефон	phone	VARCHAR	20

Продовження таблиці 2.12

Ел. пошта	Email	VARCHAR	40
Розр. рахунок	Curr_acc	VARCHAR	50
МФО	Mfo	VARCHAR	20
ЄДРПОУ	edrpou	VARCHAR	20

Таблиця 2.13. Готова продукція [створено автором]

Логічна назва поля	Фізична назва поля	Тип	Довжина поля
Код товару	prod_id	INT	Ціле
Тип товару	Prod_type	VARCHAR	50
Назва товару	prod_name	VARCHAR	80
Ціна	Price	decimal	3 подвійною точністю
В наявності	Storage_qty	VARCHAR	15
Одиниці виміру	Mesa_unit	VARCHAR	15

Таблиця 2.14. Постачальники [створено автором]

Логічна назва поля	Фізична назва поля	Тип	Довжина поля
Код постачальника	suppl_id	INT	Ціле
ПІБ постачальника	Cust_name	VARCHAR	100
ПІБ директора	Dir_name	VARCHAR	100
Адреса	Addr	VARCHAR	100
Індекс	Ind	VARCHAR	15
Телефон	phone	VARCHAR	20
Ел. пошта	Email	VARCHAR	40
Розр. рахунок	Curr_acc	VARCHAR	50
МФО	Mfo	VARCHAR	20
ЄДРПОУ	edrpou	VARCHAR	20

Таблиця 2.15. Матеріали [створено автором]

Логічна назва поля	Фізична назва поля	Тип	Довжина поля
Код матеріалу	material_id	INT	Ціле
Назва товару	material_name	VARCHAR	80
Ціна	Price	decimal	3 подвійною точністю
В наявності	Storage_qty	VARCHAR	15
Одиниці виміру	Mesa_unit	VARCHAR	15

Таблиця 2.16. Співробітники [створено автором]

Логічна назва поля	Фізична назва поля	Тип	Довжина поля
Код співробітника	empl_id	INT	Ціле
Посада	position	VARCHAR	80
ПІБ	empl_name	VARCHAR	80
Телефон	phone	VARCHAR	30

Таблиця 2.17. Маршрутний лист [створено автором]

Логічна назва поля	Фізична назва поля	Тип	Довжина поля
Код маршруту	Route_id	INT	Ціле
Назва маршруту	Route_name	VARCHAR	25

2.4.3 Кодування

Кодування представляє собою важливий етап у розробці програмного забезпечення, і це процес написання програмного коду з використанням вибраних мов програмування. Основна мета цього процесу полягає в реалізації конкретного алгоритму чи алгоритмів, які раніше були визначені. У більш докладному розумінні, кодування включає в себе перетворення моделей проектування в конкретні реалізації за допомогою обраної мови програмування.

В контексті використання Delphi, програмний код використовується для обробки подій, що виникають під час взаємодії користувача з компонентами форми. Цей етап розробки визначає, як саме буде взаємодіяти програма з користувачем через обрані компоненти, забезпечуючи відповідне функціональне виконання задач та алгоритмів.

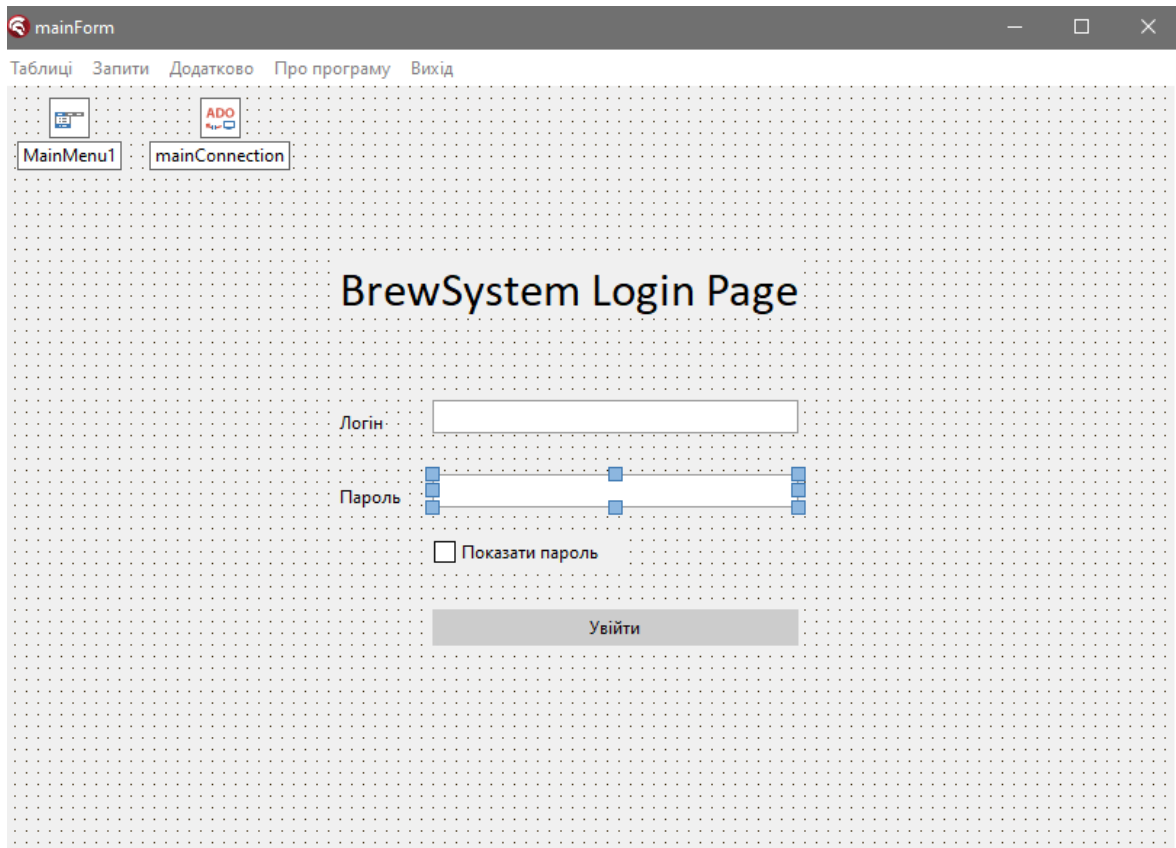


Рисунок 2.9. Форма входу [створено автором]

Приклад обробки натиснення на кнопку «Вхід» на мові програмування Delphi.

```
procedure TmainForm.Button1Click(Sender: TObject);  
var  
    i:integer;  
begin  
    if (editLog.Text <> 'admin') and (editPass.Text <> 'admin') then  
        showMessage('Логін або пароль невірний')  
    else  
        begin
```

```
for i := 0 to ControlCount - 1 do
    if (Controls[i] is TButton) or (Controls[i] is TLabel) or (Controls[i] is Tedit)
    or (Controls[i] is Tcheckbox) then Controls[i].Visible:=false;
Logo.Visible:= true;
Logo.Caption:= 'Вітаємо в системі';
mainmenu1.Items[0].Enabled:=true;
mainmenu1.Items[1].Enabled:=true;
mainmenu1.Items[2].Enabled:=true;
mainmenu1.Items[3].Enabled:=true;
mainmenu1.Items[4].Enabled:=true;
end;
end;
```

2.4.4 Тестування та налагодження програми

Тестування програмного забезпечення – це технічний процес, спрямований на вивчення якості продукту у відповідності до його контексту використання. Ця процедура включає в себе виявлення інформації щодо відповідності продукту заявленим вимогам та його функціональності, проведення пошуку помилок і оцінку програмних компонентів.

Тестування програмного забезпечення визначається як перевірка відповідності вимог, оцінка реалізованої функціональності, а також виявлення дефектів шляхом спостереження за його роботою в умовах, які моделюють різні ситуації та використання обмеженого набору тестів.

Об'єктом випробувань є програмне забезпечення для інформаційної системи складу пивного господарства. Тестування здійснюється за рядом критеріїв, включаючи відповідність вимогам, правильну реакцію на різні вхідні дані, виконання функцій у прийнятний час, практичність, сумісність з іншим програмним забезпеченням та операційними системами, а також відповідність завданням замовника.

У зв'язку з тим, що число можливих тестів навіть для нескладних програмних компонент є практично нескінченним, стратегія тестування базується на проведенні всіх можливих тестів з урахуванням доступного часу та ресурсів. Результати тестування визначаються ефективністю роботи програми та виявленням помилок чи дефектів. У процесі тестування оцінюються вхідні дані, їх обробка програмою та виведення до бази даних, щоб забезпечити працездатність модулів.

Таблиця 2.18. Виділені класи еквівалентності [Створено автором]

№ теста	Вхідні дані	Вірні класи	Невірні класи
1	Логін	admin	root
		admin	user
2	Пароль	admin	1234567890
		admin	Password

Результати тестування на основі поданих вище класів еквівалентності можна побачити в таблиці 2.19

Таблиця 2.19 Тести вірних та невірних класів еквівалентності [Створено автором]

№ теста	Класи	Вхідні дані	Вихідні дані
1	Вірний клас	Admin/Admin	Надання доступу до програми
2	Невірний клас	Admin/root	Неправильний логін/пароль
3		Admin/1234567890	Неправильний логін/пароль
4		Admin/password	Неправильний логін/пароль

Для інших модулів програми також використано аналогічні класи еквівалентності. Після проведення тестування та налагодження отримано програмний продукт, який не лише здійснює виконання запланованих функцій, але й виявляє та виправляє помилки користувача.

Висновок

На початку розділу було розглянуто теоретичні передумови моделювання інформаційної системи. Для наглядного прикладу було використано задачу

комівояжера, яку можна інтегрувати в складське господарство пивзаводу як пошук оптимального шляху з доставки сировини або продукції. Було розроблено модель системи, визначено основні потоки, створено концептуальну модель, діаграму станів програми, логічну та фізичну модель даних. Було описано фізичні та логічні таблиці бази даних. Розроблено архітектуру програми, описано модулі програмного забезпечення. Показано алгоритм роботи програми та обґрунтовані засоби для розробки програмного забезпечення інформаційної системи. Показано процес кодування та тестування ПЗ.

3. РЕАЛІЗАЦІЯ ІС СКЛАДСЬКОГО ГОСПОДАРСТВА ПИВЗАВОДУ

В результаті розробки та проектування інформаційної системи було створено базу даних та програмне забезпечення, яке дозволяє ефективно взаємодіяти користувачам з системою.

Користувачу надано інструментарій автоматизації ручних операцій, наприклад, можливість швидкого пошуку даних за критеріями. При формуванні масивів даних для завантаження в БД проводиться контроль повноти даних контроль унікальності значень ключового поля, надано систему підказок при формуванні даних, що значно полегшує процес завантаження даних.

При завантаженні програми активізується форма доступу. Меню вгорі заблоковано до тих пір, поки не буде введено правильний логін та пароль. Варто зазначити що не всі користувачі мають однаковий доступ до всіх модулів програми, так наприклад логістам доступний лише функціонал з створення накладних, їх друку та перегляд звітів, коли менеджер в свою чергу має повний доступ. Під час входу в систему визначається певний користувач, для кожного з них існують унікальні логін та пароль. На рисунку знизу можна бачити, що меню заблоковано, воно стає доступним після вводу коректних даних в поля логіну та паролю.

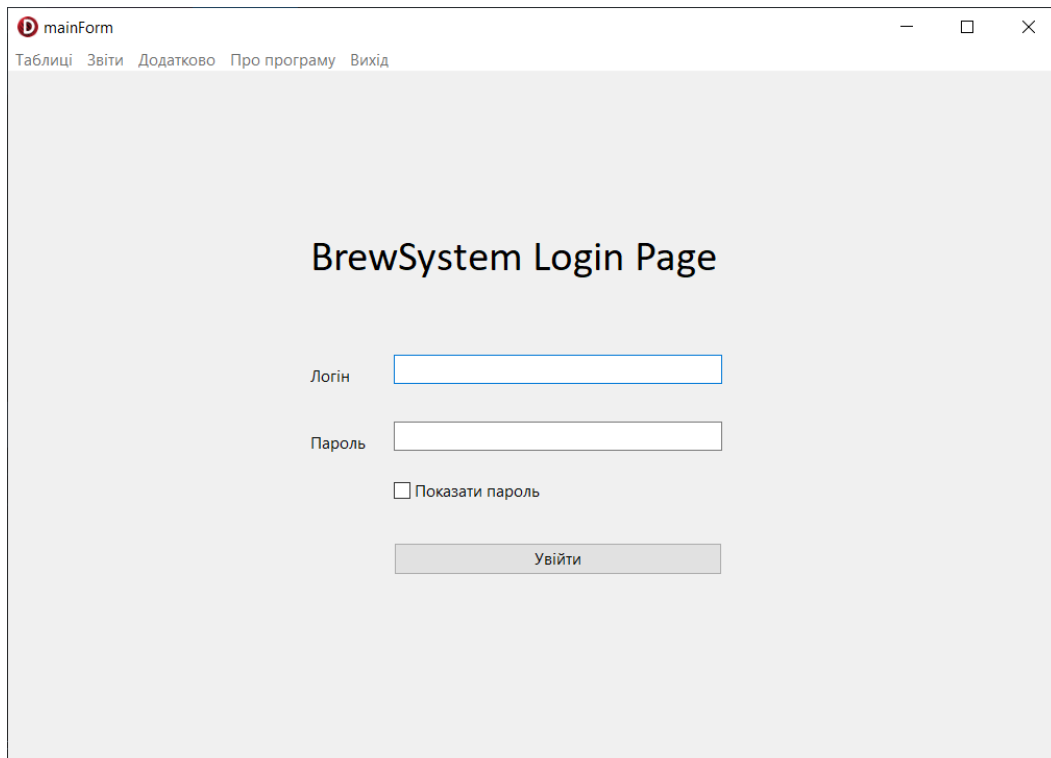


Рисунок 3.1. Форма входу [створено автором]

Якщо введений логін/пароль не було знайдено в системі, то користувач отримає помилку.

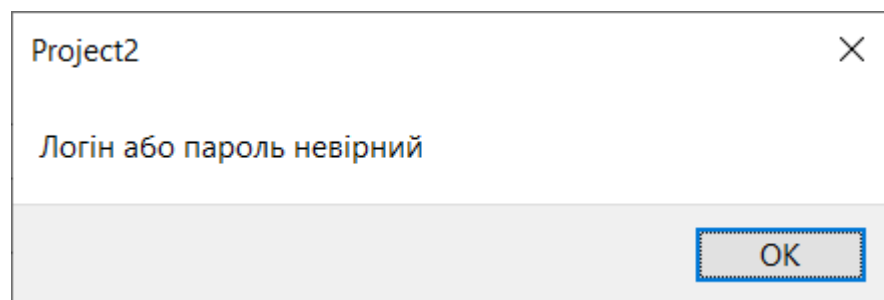


Рисунок 3.2. помилка входу [створено автором]

Також користувач може активувати відображення паролю натиснувши «галочку» відображати пароль, коли вона активна, символи в полі відображаються, в іншому випадку можна бачити спец символ, який робить пароль невидимим.

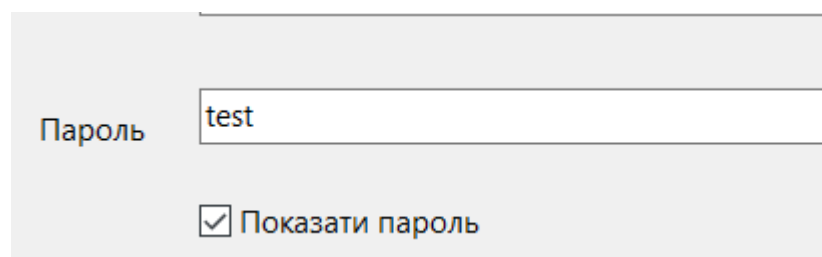
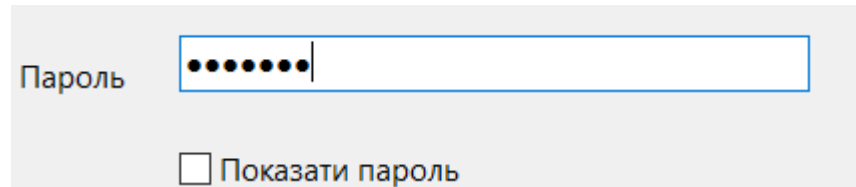


Рисунок 3.3. Відображення паролю [створено автором]



Пароль

☐ Показати пароль

Рисунок 3.4. Спец символи пароллю [створено автором]

При успішному вході користувач потрапляє на форму роботи, навігація здійснюється за допомогою меню зверху.

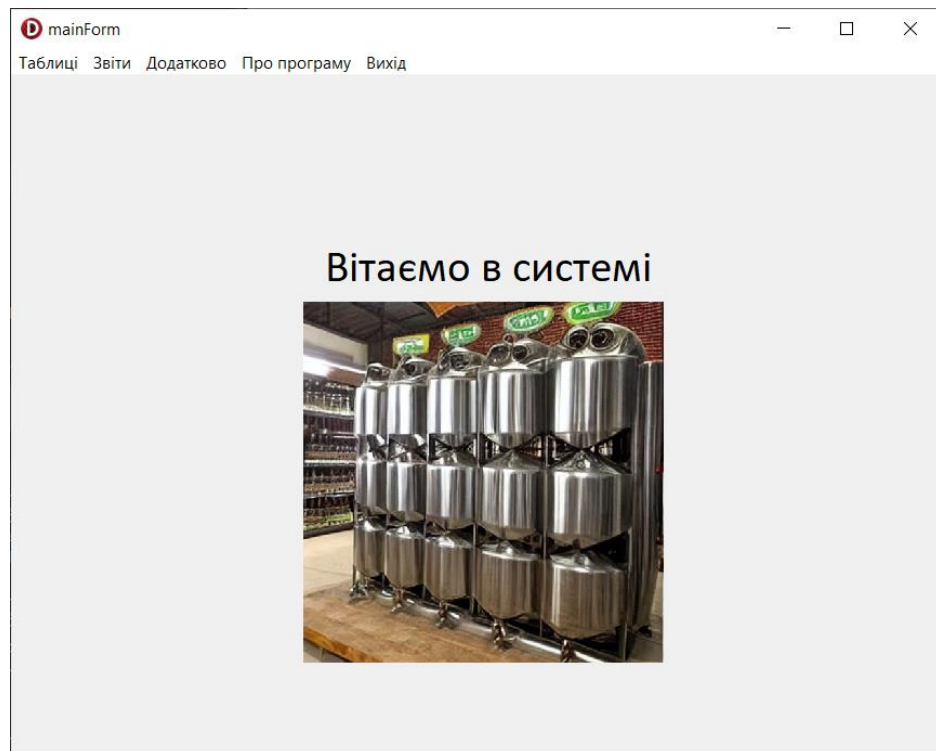


Рисунок 3.5. Форма після входу [створено автором]

При виборі таблиці користувач потрапляє на відповідну форму, де може редагувати/додавати/видаляти записи бази даних. Розглянемо детальніше на прикладі таблиці сировина

materialsForm

Пошук

Очистити поле пошуку

ID

4

Додати запис

Назва сировини

Хміль

Зберегти

Ціна

90

Видалити

К-ть на складі

500

Одиниці виміру

kg

mate	material_name	price	storage_	meas_unit
1	Солод ячмінний	60	500	kg
2	Солод пшеничний	75	500	kg
3	Солод житній	80	500	kg
4	Хміль	90	500	kg
5	Дріжді	55	500	kg

⏪

⏩

Рисунок 3.6. Приклад форми [створено автором]

Під час активізації форми користувач бачить зверху, меню пошуку, в даному випадку пошук відбувається по всім доступним полям таблиці. Для пошуку потрібно почати вводити символи в поле пошуку, кнопка «Очистити поле пошуку» очищає відповідне поле. На рисунку нижче можна бачити результати пошуку

materialsForm

— □ ×

Пошук

солод

Очистити поле пошуку

ID

1

Додати запис

Назва сировини

Солод ячмінний

Зберегти

Ціна

60

Видалити

К-ть на складі

500

Одиниці виміру

kg

mate	material_name	price	storage_	meas_unit
► 1	Солод ячмінний	60	500	kg
2	Солод пшеничний	75	500	kg
3	Солод житній	80	500	kg

⏪

⏴

⏵

⏩

Рисунок 3.7. Приклад пошуку [створено автором]

Кнопка «Додати запис» додає новий запис в таблицю, він буде пустий доти, доки користувач не заповнить всі поля, в даному випадку їх 4, поле «ІД» заповнюється автоматично.

Як тільки всі поля будуть заповнені користувач повинен натиснути кнопку «Зберегти», після чого запис буде остаточно додано в таблицю.

materialsForm

Пошук

солод

Очистити поле пошуку

ID

6

Додати запис

Назва сировини

test

Зберегти

Ціна

10,1

Видалити

К-ть на складі

11

Одиниці виміру

test

mate	material_name	price	storage_	meas_unit
1	Солод ячмінний	60	500	kg
2	Солод пшеничний	75	500	kg
3	Солод житній	80	500	kg
6	test	10,1	11	test

⏪

⏴

⏵

⏩

Рисунок 3.8. Доданий запис [створено автором]

Якщо користувач при додаванні запису залишить одне або декілька полів в не заповненому стані, то пр натисненні на кнопку «Зберегти» буде отримано насупне повідомлення, та дані не будуть збережені.

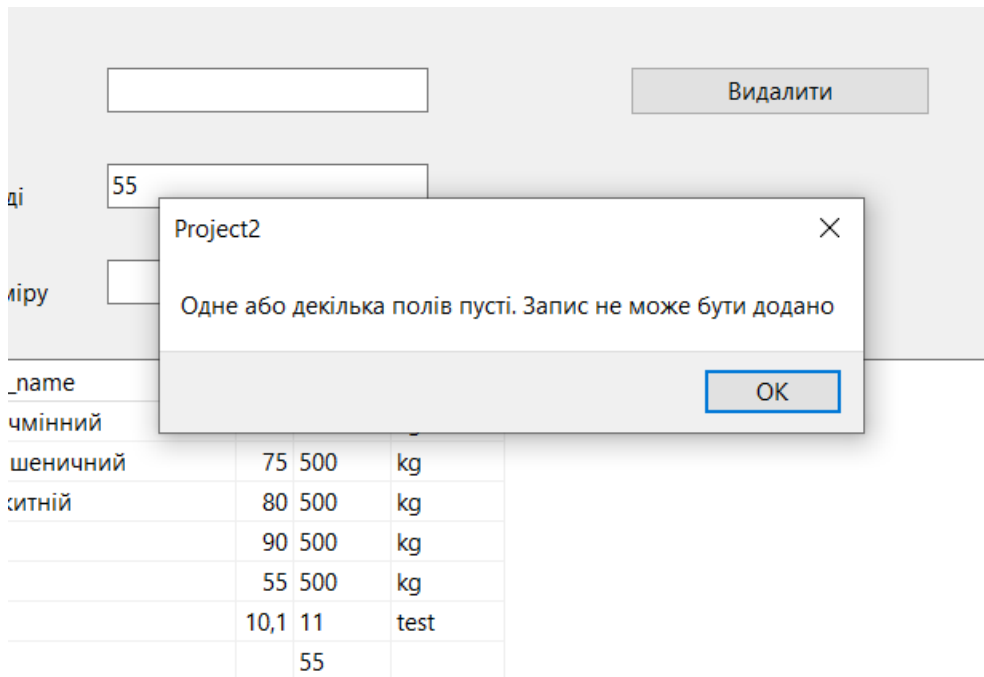


Рисунок 3.9. Помилка при додаванні запису [створено автором]

Також користувач може видалити запис з таблиці, для цього необхідно виділити потрібне поле запису, після чого натиснути кнопку «видалити». Під час видалення у користувача буде запитано чи хоче він видалити запис.

Для навігації по таблиці знизу розташований компонент «Навігатор», при натисканні кнопок користувач переміщується по одному запису або відразу в початок/кінець таблиці.



Рисунок 3.10. Навігація по таблиці [створено автором]

Однією з найважливіших функцій програми є створення та друк накладних, даний функціонал реалізовно на формі накладних.

Пошук

Очистити поле пошуку

Роздрукувати накладну

ID

1

Од. виміру

kg

Співробітник

Іванов Іван Іванович

Вага

123

Тип маршруту

Відправлення

Дата створення накладної

21.11.2023

Замовник

Іван Іванов

Дата транспортування

21.11.2023

Продукція

Чернігівське

Додати запис

Зберегти

Видалити

inv_id

empl_id_name

route_id_name

cust_id_name

prod_id_name

meas_unit

weight

inv_creation_date

trans_date

1

Іванов Іван Іванович

Відправлення

Іван Іванов

Чернігівське

kg

123

21.11.2023

21.11.2023

2

Сидоренко Марія Сергіївна

Відправлення

Сергій Сидоров

Хугарден

kg

55

25.11.2023

25.11.2023

5

Костенко Олександр Володимирович

Відправлення

Олена Степаненко

Чернігівське

kg

12

12.12.2023

12.12.2023

Режим роботи

●

Накладні готової продукції

○

Накладні сировини

Рисунок 3.11. Таблиця накладна [створено автором]

Для друку накладної треба обрати запис в таблиці та натиснути кнопку «роздрукувати накладну». Після чого відкриється форма з створеною накладною, яку можна роздрукувати або експортувати в певний файл, наприклад PDF або docx для перегляду в інших додатках та його передавання іншими методами.

The screenshot shows a window titled "printInvForm" with a menu bar containing "Експорт" and "Роздрукувати". The main area displays a receipt form titled "Накладна на Відправлення номер 1". The form contains the following fields:

Field	Value
Співробітник	Іванов Іван Іванович
Замовник	Іван Іванов
Продукт	Чернігівське
Одиниці виміру	kg
Кількість	123
Дата створення накладної	21.11.2023
Дата транспортування	21.11.2023
Роздруковано	10.12.2023 15:15:10
Підпис	

Рисунок 3.12. Створена накладна [створено автором]

При натисненні кнопки «роздрукувати» відкриється стандартне вікно принтерів Windows, де можна обрати параметри друку.

The screenshot shows the standard Windows "Print" dialog box. It includes the following sections and options:

- Printer:** Name: Microsoft Print to PDF (with a Properties... button).
- Where:** Print to file (checkbox).
- Pages:** All (selected), Current page, Pages: (text field). Below: Enter page numbers and/or page ranges, separated by commas. For example, 1,3,5-12.
- Copies:** Number of copies: 1 (spinner), Collate (checked checkbox).
- Other:** Print: All pages (dropdown), Order: Direct (1-9) (dropdown), Duplex: Default (dropdown).
- Print mode:** 1 > 1 Default (dropdown).
- Print on sheet:** Default (dropdown).
- Buttons:** OK, Cancel.

Рисунок 3.13. Вікно друку [створено автором]

Для виходу з програми необхідно натиснути кнопку «Вихід», після чого відобразиться вікно з підтвердженням виходу

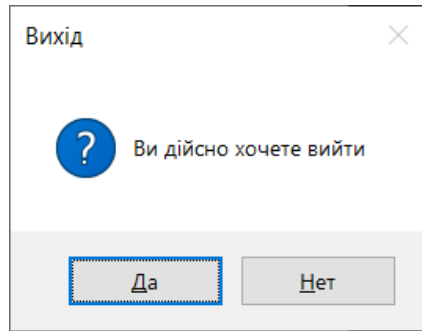


Рисунок 3.14. Вихід з програми [створено автором]

Висновок

В даному розділі описано програмний продукт, який було розроблено для інформаційної системи пивзаводу. Описано керування та пояснено основні принципи роботи ПЗ. Завдяки ньому користувач може взаємодіяти з системою через інтерфейс програми. Вносити дані в базу даних використовуючи створені форми та друкувати накладні зі звітами.

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВПРОВАДЖЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ СКЛАДСЬКОГО ГОСПОДАРТСВА ПИВЗАВОДУ

4.1 Загальні положення

На сьогоднішній день сучасні обчислювальні засоби широко використовуються в різних аспектах повсякденної людської діяльності. Вони вирішують різноманітні завдання, від автоматизації монотонної ручної праці до вирішення складних технічних розрахунків, які вимагають колективної наукової роботи. Засоби програмування на сьогоднішній день дозволяють ефективно вирішувати завдання, що раніше потребували значних зусиль від наукових співробітників.

Розширення нових технологій та зростання застосування обчислювальної техніки призвело до широкого використання електронних обчислювальних систем для рішення завдань, пов'язаних з обробкою, контролем і аналізом інформації. Розробка та впровадження інтегрованих інформаційних систем, зокрема в галузі управління, проектування та технологічної підготовки виробництва, є ключовою в сучасному підході до оптимізації виробничих процесів.

Успішне впровадження таких систем та їхня роль у розвитку національного господарства значною мірою залежать від професіоналізму фахівців з системотехніки. Знання методів аналізу та проектування таких систем, володіння можливостями обчислювальної техніки і математичними методами стає невід'ємним фактором успіху в цій галузі.

ІС складського господарства пивзаводу призначена для автоматизації завдань працівників складу. Інтеграція цієї системи може призвести до значної економії, вираженої в підвищенні продуктивності праці та вивільненні робочого часу.

Однак з точки зору розробника, важливим аспектом є процес формування вартості системи, яка є унікальним товаром зі своїми специфічними особливостями. Створення ІС передбачає одноразові витрати на її розробку та

придбання необхідного обладнання, а також поточні витрати на її функціонування. Економія, яку надає функціонування такої системи, оцінюється врахуванням витрат на експлуатацію, і її ефективність визначається відношенням цієї економії до витрат на створення системи. Усі економічні показники розраховуються за діючими оптовими цінами, тарифами та заробітними платами.

4.2 Розрахунок вартості матеріально – енергетичних витрат

Кошторис виробництва охоплює витрати підприємства, пов'язані з його основною діяльністю за конкретний період, незалежно від того, чи вони включаються в собівартість продукції протягом цього періоду чи ні. Отже, кошторис виробництва та собівартість загального обсягу продукції, як правило, не збігаються. Складання кошторису виробництва здійснюється за різними економічними елементами.

Матеріальні витрати, як складова частина кошторису, включають витрати на сировину та основні матеріали, що становлять матеріальну основу продукції; вироби, які необхідно придбати для комплектації продукції (двигуни, прилади і т.д.); придбані напівфабрикати (штампування, відливання, ковані вироби і т.д.); виробничі послуги сторонніх підприємств та організацій, які необхідні для виробництва продукції; допоміжні матеріали, що використовуються в технологічному процесі (кріпильні деталі, фарби, інструменти і т.п.) або необхідні для його обслуговування (ремонт, експлуатація устаткування і т.д.), а також на господарські та управлінські потреби (утримання будівель, канцелярські товари і т.п.); паливо та енергію (електроенергію, пар, газ і т.д.) зовнішніх джерел. Витрати на виробництво енергії включаються в кошторис окремими елементами; пошук і використання природних ресурсів (витрати на геологорозвідувальні роботи, рекультивацію земель, вартість деревини і т.д.). Розрахунок витрат на матеріали виконується з врахуванням норм витрат та цін з урахуванням транспортних та заготівельних витрат, які не включаються в інші елементи кошторису (плата за транспортування, роботи з вантажоперевезення, комісійні підприємствам-

заготівельникам і т.д.). Вартість відходів віднімається від вартості матеріалів відповідно до їх використання чи продажу.

Таблиця 4.1 Розрахунок вартості матеріальних витрат [створено автором]

Найменування матеріалу	Одиниці виміру	Ціна за одиницю	Загальна вартість
Флеш - накопичувач	Штук	200, 00 грн.	400, 00 грн.
Інтернет	Мбіт/сек	400,00 грн.	400, 00 грн.
Роутер	Штук	400, 00 грн.	400, 00 грн.
Папір	Штук	1, 00 грн.	500, 00 грн.
Картриджі та тонери для принтеру	Штук	600, 00грн.	1800, 00 грн.
Література	Штук	~600 грн.	1800 грн.
Непередбачувані витрати		500 грн.	500 грн.
Всього			5800, 00 грн.

Транспортно - заготовельні витрати складають 20% від суми матеріальних витрат, тобто $5800,00 \times 20:100 = 1160,00$ грн.

Разом матеріальні витрати з ТЗВ складають:

$$Ц_{заг} = Ц_{м} + Ц_{тзр} \quad (1.3)$$

$$Ц_{заг} = 5800,00 + 1160,00 = 6960,00 \text{ грн}$$

Розрахунок вартості електричної енергії

Кількість комп'ютерів – 2 шт.

Кількість принтерів – 1 шт.

Вартість електрики розраховується за формулою:

$$В_{ел.} = N \times Д \times \gamma \times Ц \quad (1.4)$$

Де N – потужність ПК/Принтера (кВт)

Д – кількість робочих днів (35)

γ – тривалість робочої зміни (8 год)

Ц – вартість 1 кВт/год. (2,64 грн.)

$$N_{ПК} = 0,22 \times 3 = 0,66 \text{ кВт}$$

Принтер працює 3 години в день.

$$N_{\text{принт.}} = 0,55 \times 3 = 1,65 \text{ кВт}$$

Відповідно розрахуємо вартість споживання електроенергії комп'ютерами:

$$V_{\text{ЕЛ.ПК}} = 0,66 \times 35 \times 8 \times 2,64 = 487,87 \text{ грн.}$$

Вартість споживання електроенергії принтера:

$$V_{\text{ЕЛ.Принт.}} = 35 \times 1,65 \times 2,64 = 152,46 \text{ грн.}$$

Сума вартості електроенергії дорівнює:

$$487,87 \text{ грн.} + 152,46 \text{ грн.} = 640,33 \text{ грн.}$$

Загальна вартість матеріально – енергетичних витрат становить:

$$6960 \text{ грн.} + 640,33 \text{ грн.} = 7600,33 \text{ грн.}$$

4.3 Розрахунок фонду оплати праці за виконану роботу

Заробітна плата охоплює всі форми винагороди за труд штатного та позаштатного персоналу, зайнятого у виробництві продукції, обслуговуванні виробничого процесу та управлінні. У собівартість не включаються виплати праці, фінансовані із прибутку або інших джерел, що мають спеціальне призначення.

Згідно із Законом України "Про оплату праці", заробітна плата є грошовою винагородою за виконану роботу, яку власник чи уповноважений ним орган виплачує працівникові на підставі трудового договору. Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійних якостей працівника, результатів його праці та господарської діяльності підприємства.

У ринковій економіці заробітна плата виконує кілька функцій, зокрема відтворювальну, стимулюючу, регулюючу та соціальну. Заробітна плата складається з основної та додаткової частин, а також заохочувальних та компенсаційних виплат.

Основна заробітна плата представлена винагородою за роботу в межах норм праці (норми часу, виробітку, обслуговування, посадові обов'язки) і визначається тарифними ставками для робітників і посадовими окладами для службовців.

Додаткова зарплата включає винагороду за працю понад встановлені норми, трудові успіхи, винахідливість і особливі умови праці, охоплюючи доплати, надбавки, гарантії та компенсаційні виплати, передбачені законодавством.

Інші компенсаційні та заохочувальні виплати включають винагороду за річну роботу, премії за спеціальними системами і положеннями, а також інші грошові й матеріальні виплати, які не визначені законодавством або проводяться понад встановлені норми.

Мінімальна заробітна плата, законодавчо визначений розмір оплати за просту, некваліфіковану працю, визначає державну соціальну гарантію і обов'язкова для всіх підприємств незалежно від форми власності та господарювання.

1. Трудомісткість (Т) - показник, що характеризує витрати робочого часу на виробництво певної споживної вартості або на виконання конкретної технологічної операції.

Трудомісткість становить 810 годин.

2. $C_{ч1/2/3/сер.роб}$ – середня годинна тарифна ставка, встановлюється керуючим, в залежності від професійності працівників.

$$C_{ч1.роб} = 110,00 \text{ грн.}$$

$$C_{ч2.роб} = 99,00 \text{ грн.}$$

$$C_{ч3.роб} = 100,00 \text{ грн.}$$

$$C_{ч.сер} = \frac{C_{ч1.роб} + C_{ч2.роб} + C_{ч3.роб}}{3} \quad (1.5)$$

$$C_{ч.сер} = \frac{110,0 + 99,0 + 100,0}{3} = 103,00 \text{ грн.}$$

3. Фонд оплати праці (ФОП) - основний економічний показник з оплати праці, що широко використовується у господарській практиці, статистичній звітності, економічному аналізі.

$$ФОП = Z_{осн.} + Z_{дод.} \quad (1.6)$$

Де: $Z_{осн}$ – основна заробітна плата.

$$Z_{осн} = C_{ч.сер} \times T \quad (1.7)$$

$$Z_{осн} = 103,0 \times 810 = 83\,430,00 \text{ грн.}$$

$Z_{\text{дод.}}$ – додаткова заробітна плата, (премія) що становить 25 – 40% від основної заробітної плати, в нашому випадку премія – 27%.

$$Z_{\text{дод.}} = \frac{\Phi_{\text{прем}} * Z_{\text{осн}}}{100} \quad (1.8)$$

$$Z_{\text{дод.}} = \frac{27 * 83430,0}{100} = 22526,1 \text{ грн.}$$

$$\text{ФОП} = 83\,430,00 + 22526,1 = 105\,956,1 \text{ грн.}$$

4. Єдиний соціальний внесок (ЄСВ) - обов'язковий платіж до системи загальнообов'язкового державного соціального страхування, що справляється в Україні з метою забезпечення страхових виплат за поточними видами загальнообов'язкового державного соціального страхування (становить 22% від ФОП).

$$\text{ЄСВ} = \frac{22 * \text{ФОП}}{100} \quad (1.9)$$

$$\text{ЄСВ} = \frac{22 * 105\,956,1}{100} = 23310,342 \text{ грн.}$$

5. $Z_{\text{сер.}}$ - Середня заробітна плата одного працівника за день

$$Z_{\text{сер.}} = \frac{\text{ФОП}}{D * N}$$

$$Z_{\text{сер.}} = \frac{105\,956,1}{35 * 3} = 1009,106 \text{ грн.}$$

N - Число робітників – 3

D - Тривалість виконання – 35 днів

$Z_{\text{сер.міс}}$ – середня зарплата одного працівника за місяць

$$Z_{\text{сер.міс}} = Z_{\text{сер.}} \times 22$$

$$Z_{\text{сер.міс}} = 1009,106 \times 22 = 22\,200,332 \text{ грн.}$$

4.4 Розрахунок ціни розробки

Собівартість продукції представляє собою сукупні витрати, виражені у грошовій формі, які виникають при підготовці та випуску продукції. У більш загальному контексті собівартість можна визначити як грошовий вираз обсягу використаних ресурсів з конкретною метою. Ця концепція собівартості включає три ключові аспекти:

1. Собівартість відображає, скільки і які ресурси було використано в процесі виробництва.

2. Величина використаних ресурсів виражена у грошовому виразі, що дозволяє розраховувати загальну вартість цих ресурсів.

3. Конкретна мета використання ресурсів визначає об'єкт собівартості, будь то виробництво або реалізація.

Таблиця 4.2 Розрахунок ціни розробки програмного забезпечення [створено автором]

Статті калькуляції	Методика розрахунку	Сума в гривнях
1) Основні і додаткові матеріали	За розрахунками в пункті 5.1	5800,00
2) Транспортно -заготівельні витрати	За розрахунками в пункті 5.1	1160,00
3) Енергетичні витрати	За розрахунками в пункті 5.1	640,33
4) Основна заробітна плата	$Z_{\text{осн}} = C_{\text{ч.сер}} \times T$	83 430, 00
5) Додаткова заробітна плата	27% від основної заробітної плати	22526, 1
6) ЄСВ	22% від ст. 3+4	23310, 342
7) Амортизація відрахувань	За розрахунком кумулятивним методом	2620,36
8) Загальні виробничі витрати	90% від основної заробітної плати	75087
9) Виробнича собівартість	Сума статей 1 - 8	214 574,132
10) Адміністративні витрати	73% від основної заробітної плати	60 903,9

Продовження таблиці 4.2

11) Повна виробнича собівартість	Сума статей 9-10	275 478,032
12) Рівень рентабельності	31% від статті 11	85 398,19
13) Ціна розробки	Сума статей 11 – 12	360 876,222

Використовуючи кумулятивний метод, річну суму амортизації визначають як добуток вартості, яка амортизується, та кумулятивного коефіцієнта. Кумулятивний коефіцієнт розраховується діленням кількості років, що залишаються до кінця терміну корисного використання об'єкта основних засобів, на суму кількості років його корисного використання.

Кумулятивний метод належить до групи прискорених методів амортизації. Отже, всі принципові особливості цієї групи подібні та полягають у тому, що значна частка амортизаційних відрахувань накопичується протягом перших років експлуатації. Тому кумулятивний метод застосовують у тих самих випадках, що й інші два методи прискореної групи.

Кількість комп'ютерів – 2 шт.

Кількість принтерів – 1 шт. (не враховується)

Вартість комп'ютера – 15000, 0 грн.

Ліквідаційна вартість комп'ютера – 500, 0 грн.

Враховуючи те, що комп'ютер використовується перший рік:

$$BA = BP - BL \quad (1.10)$$

BA – вартість, яка амортизується

BP – первісна вартість

BL – ліквідаційна вартість

$K_{\text{кум 1 рік}}$ – кумулятивний коефіцієнт

$$K_{\text{кум 1 рік}} = \frac{2}{1+2} = \frac{2}{3} = 0,67$$

$$BA = 2 \times 15000,00 - 500,00 = 29500,00 \text{ грн.}$$

A - сума амортизаційних відрахувань

$$A_{\text{рік}} = 29500,00 \times 0,67 = 19765,00 \text{ грн.}$$

$$A_{\text{міс}} = 19765,00 : 12 = 1647,08 \text{ грн.}$$

$$A_{\text{день}} = A_{\text{міс.}} : 22$$

$$A_{\text{день}} = 1647,08 : 22 = 74,87 \text{ грн.}$$

$$A_{\text{заг.}} = A_{\text{день}} \times 35$$

$$A_{\text{заг.}} = 74,87 \times 35 = 2620,36 \text{ грн.}$$

4.5 Розрахунок економічної ефективності та строку окупності розробленої ІС

Ефективність - це економічний термін, який відображає співвідношення між результатами, отриманими внаслідок певної діяльності, і ресурсами, які були витрачені для їх досягнення. При вимірюванні ефективності можна враховувати ресурси як за їхньою первісною вартістю (використані ресурси), так і частковою вартістю у формі виробничих витрат (виробничо спожиті ресурси). Оскільки результати виробництва можуть бути представлені в різних формах, таких як вартісна, натуральна, соціальна, важливо визначити ефективність відповідно до аспектів діяльності підприємства, які потребують аналізу та оцінки.

Період окупності проекту визначає час, необхідний для того, щоб сума отриманих коштів від реалізації проекту компенсувала витрати на його впровадження. Цей період зазвичай вимірюється у роках або місяцях.

Економічна ефективність розраховується за формулою:

$$ЕФ = \Delta C - K_{\text{еф.}} * BC \quad (1.11)$$

Де: ΔC – вивільнені кошти після впровадження системи

$K_{\text{еф.}}$ – коефіцієнт економічної ефективності, що становить 0,6

BC – виробнича собівартість (з табл. №3 дорівнює 214 574,132 грн.)

$$\Delta C = \Phi ЗП_{1\text{рік}} - EB \quad (1.12)$$

Де: $\Phi ЗП_{1\text{рік}}$ – сума коштів, яка вивільнюється на підприємстві після впровадження ІС

$$EB = A_{\text{відр.}} + MB + B_{\text{ел.}} \quad (1.13)$$

Де: $A_{\text{відр.}}$ – сума амортизаційних відрахувань (становить 2620,36 грн.)

MB – вартість матеріальних витрат (становить 6960 грн.)

$V_{\text{ел.}}$ – вартість електроенергії (становить 640,33 грн.)

$$EB = 2620,36 + 6960,00 + 640,33 = 10\,220,69 \text{ грн.}$$

Пряму економічну ефективність визначають ґрунтуючись на тому, що за експертною оцінкою фахівців підприємства впровадження інформаційної системи вивільняє одного працівника, середня зарплата якого складає:

$$\Phi ЗП_{\text{м}} = 13000,00 \text{ грн./міс}$$

Вивільнені кошти за рік складають:

$$\Phi ЗП_{\text{1 рік}} = \Phi ЗП_{\text{м}} \times 12 \quad (1.14)$$

$$\Phi ЗП_{\text{1 рік}} = 13000,00 \times 12 = 156\,000,0 \text{ грн.}$$

Вивільнені кошти після впровадження за один рік становлять 156 000, 00 грн.

Отже у перший рік витрати на створення і експлуатацію ІС складуть:

$$\Delta C = 156\,000,00 - 10\,220,69 = 145\,779,31 \text{ грн.}$$

Строк окупності системи визначається як відношення витрат на розробку продукту до вивільнених коштів від впровадження цього продукту:

$$T_{\text{ок}} = BC : \Delta C$$

$$T_{\text{ок}} = 214\,574,132 : 145\,779,31 = 1,5 \text{ роки}$$

Отже, термін окупності інформаційної системи складського господарства пивного заводу становить 1,47 роки.

Висновок

В даному розділі було економічно обґрунтовано впровадження та розробка інформаційної системи.

Розраховано вартість матеріально енергетичних витрат та фонду оплати праці за виконану роботу для персоналу. Визначено такі поняття ЄСВ, ФОП, нормо-години, трудомісткість. Розраховано ціну розробки інформаційної системи та строк окупності її окупності.

5 ОХОРОНА ПРАЦІ

5.1 Вступ

Згідно зі статтею 1 Закону України "Про охорону праці", термін "охорона праці" визначається як система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів і засобів, спрямованих на збереження здоров'я і працездатності людини під час виконання праці. Конкретні заходи, спрямовані на створення безпечних та здорових умов праці, попередження нещасних випадків і професійних захворювань, регулюються спеціальними правилами та нормами відповідно до вимог Кодексу законів про працю. Закон України "Про охорону праці" визначає, що нормативні акти можуть бути загальними для всіх галузей економіки або специфічними для певної галузі. Державні міжгалузеві та галузеві нормативні акти про охорону праці охоплюють правила, стандарти, норми, положення, інструкції та інші документи, які мають юридичну силу і є обов'язковими для виконання.

5.2 Аналіз шкідливих та небезпечних факторів

Фактори, що становлять небезпеку і шкоду на робочому місці програміста, можна поділити за природою виникнення на кілька груп:

- Фізичні;
- Хімічні;
- Психофізіологічні;
- Біологічні.

В робочому приміщенні програміста можуть негативно впливати наступні фізичні фактори:

- Зміни в температурі повітря;
- Висока чи низька вологість повітря;
- Недостатня освітленість робочого місця;
- Шум, що перевищує припустимі норми;
- Підвищений рівень іонізуючого випромінювання;

- Підвищений рівень електромагнітних полів;
- Підвищений рівень статичної електрики;
- Небезпека поразки електричним струмом;
- Блик екрана дисплея.

Щодо хімічно небезпечних факторів, які постійно впливають на програміста, можна виділити виникнення активних часток при роботі комп'ютера.

Біологічні шкідливі фактори відсутні в даному приміщенні.

До психологічно шкідливих факторів, які впливають на програміста протягом його робочої зміни, можна віднести:

- Нервово-емоційне перевантаження;
- Розумова напруга;
- Перенапруга зорового аналізатора.

Детальніше розглянемо небезпечні і шкідливі фактори, що виникають у зв'язку з розробкою даної системи.

Суттєвим фізичним фактором є мікроклімат робочої зони, зокрема температура й вологість повітря. Висока температура у поєднанні з високою вологістю може значно вплинути на працездатність користувача, збільшуючи час реакції, порушуючи координацію рухів та збільшуючи кількість помилкових дій. Висока температура також негативно впливає на психологічні функції, знижуючи увагу, обсяг оперативної пам'яті та здатність до асоціацій.

В офісних приміщеннях часто спостерігається низька вологість повітря, що може призводити до сухості шкіри, особистих захворювань шкіри та почуття спраги.

Температура, вологість та швидкість руху повітря взаємодіють на теплообмін, і їх слід розглядати як комплексний вплив. Порушення теплообміну може призводити до теплового перегріву зі схожими наслідками, такими як висока температура тіла, потовиділення, прискорений пульс[13].

Таблиця 5.1 Оптимальні норми температури, відносної вологості й швидкості руху повітря.

Період року	Категорія праці	Температура	Відносна вологість повітря (%)	Швидкість руху повітря, не більше (м/с)
Холодний і перехідний	Легка	20-23	60-40	0.2
Теплий	легка	22-25	60-40	0.2

5.3 Розрахунок системи штучного освітлення приміщення відділу інформаційних технологій

Забезпечення належних умов освітлення на робочому місці визнається ключовим фактором для створення комфортних умов праці, особливо важливим при використанні моніторів, оскільки очі користувача найбільше напружуються під час роботи з ними.

При організації освітлення необхідно враховувати, що збільшення рівня освітленості може призвести до зменшення контрасту на екрані. Таким чином, при виборі джерел загального освітлення важливо враховувати яскравість та спектральний склад випромінювання.

Загальна чутливість зорової системи зростає разом із збільшенням рівня освітленості в приміщенні, але це має місце лише до певного рівня, після чого додаткове збільшення може призвести до значного зменшення контрасту.

Для визначення оптимального рівня освітленості важливо:

1. Визначити необхідний рівень освітленості для обличчя дисплеїв, враховуючи зовнішні джерела світла.

2. У випадках, коли необхідний рівень освітленості не є прийнятним для інших працівників, забезпечити збереження необхідного контрасту за допомогою інших засобів.

Рекомендовані співвідношення яскравості в полі зору:

- Між екраном і документом: 1:5 - 1:10.
- Між екраном і поверхнею робочого столу: 1:5.
- Між екраном і клавіатурою, а також між клавіатурою і документом: не більше 1:3.

- Між екраном і навколишніми поверхнями: 1:3 - 1:10.

Місцеве освітлення на робочому місці оператора здійснюється світильником, розташованим на робочому столі. Важливо, щоб він мав непросвічуючий відбивач і розташовувався нижче лінії зору оператора, уникнути осліплення.

Робоче місце знаходиться поруч із вікном, тому екран монітора розташовується під прямим кутом до нього, для уникнення відблисків на екрані.

Для уникнення відбиттів, які можуть знизити чіткість зображення, робоче місце не розташовується прямо під джерелом верхнього світла, а трохи осторонь.

Стіни та поверхні за комп'ютером освітлені приблизно так само, як і екран. Треба уникати дуже яскравого чи блискучого фарбування на робочому місці, що може стати джерелом відбиттів, що призводять до неприємностей[14].

5.4 Розробка заходів щодо зменшення впливу шкідливих та небезпечних факторів при роботі з ПК

5.4.1 Заходи щодо зменшення впливу

Базуючись на визначених негативних факторах, було складено перелік рекомендацій щодо дотримання правил роботи з комп'ютером, який містить інформацію про заходи для забезпечення безпеки при взаємодії з комп'ютерами та електромагнітними полями:

1. Слід утримуватись відповідно до всіх вимог, які стосуються приміщень, де проводиться робота з комп'ютером.
2. Необхідно ретельно враховувати коректне розташування робочих місць як відносно конкретної кімнати, так і відносно всього будинку чи підприємства.

3. Треба подбати про те, щоб було забезпечено сучасні комп'ютерні меблі, які є зручними та не викликають відчуття дискомфорту.

4. Потрібно звернути увагу на те, щоб мати сучасне комп'ютерне обладнання (рідкокристалічні екрани, ергономічні клавіатури тощо).

5. Оскільки електромагнітне випромінювання виходить з усіх частин монітора, безпечніше розміщувати комп'ютер в куту кімнати або місці, недоступному для тих, хто не працює з ним, збоку або позаду.

6. Не залишати комп'ютер або монітор ввімкненими на тривалий період.

7. Обмежити тривалість роботи за комп'ютером без перерви, рекомендується робити короткі перерви кожні 30 хвилин.

8. Використовувати вправи для очей, які допомагають уникнути проблем, пов'язаних із використанням комп'ютера.

9. Урізноманітнити види занять за комп'ютером, наприклад, роботу з текстовим процесором можна поєднувати з переглядом відеороликів.

10. Обирати монітори з високою роздільною здатністю, оскільки вони завжди більш комфортні для очей, ніж монітори з низькою роздільною здатністю.

11. Застосовувати засоби для зменшення відблисків від монітора, оскільки яскраве та нерівне освітлення в кімнаті може викликати неприємні відображення на екрані.

5.4.2 Техніка безпеки при роботі на комп'ютері

Правила техніки безпеки для користування комп'ютером:

1. Перед початком роботи перевірте, чи в належному стані всі кабелі та розетки.

2. Не залишайте комп'ютер без нагляду, якщо він увімкнений.

3. Утримуйтеся від перетягування кабелю живлення комп'ютера.

4. Не використовуйте комп'ютер у небезпечних умовах, таких як близькість до води або відкритого вогню.

5. Утримуйтеся від вживання алкогольних напоїв або наркотичних речовин перед роботою на комп'ютері.

Специфічні правила техніки безпеки для розробників програмного забезпечення:

1. Забезпечте правильне розташування тіла. Спинка стільця повинна бути прямою, а ноги зігнуті під прямим кутом.
2. Розташуйте монітор на відстані 60-70 см від очей.
3. Робіть перерви для відпочинку кожні 20-30 хвилин.
4. Уникайте перенапруження очей.
5. Не працюйте за комп'ютером занадто довго.

Висновок

В даному розділі було розглянуто таке питання як охорона праці при розробці інформаційної системи.

Визначено термін «охорона праці» та чим вона визначається. Проаналізовано, які шкідливі та небезпечні речовини фактори можуть виникати під час роботи за комп'ютером. Було розраховано та визначено, яка температура та яке штучне освітлення повинно бути в приміщенні роботи розробника. Було розроблено заходи щодо зменшення впливу шкідливих та небезпечних факторів при роботі з ПК. Було встановлено правила безпеки під час роботи за комп'ютером

6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

6.1 Вступ

Охорона оточуючого середовища включає в себе комплекс заходів, спрямованих на збереження та раціональне використання природних ресурсів, а також захист від забруднення та деградації. Ця ініціатива є важливою складовою сталого розвитку, сприяючи створенню сприятливих умов для життя людини та збереження біорізноманіття та природних ресурсів для майбутніх поколінь.

Головна мета охорони навколишнього середовища полягає в забезпеченні сталого розвитку суспільства, створенні безпечних та комфортних умов для життя людини, а також в збереженні природного середовища для майбутніх поколінь.

Основні завдання охорони навколишнього середовища включають:

1. Захист повітря, води та ґрунтів від забруднення.
2. Контроль за використанням природних ресурсів.
3. Захист біорізноманіття.
4. Зменшення рівня забруднення оточуючого середовища.
5. Відновлення природних ресурсів.

Охорона навколишнього середовища здійснюється на різних рівнях, включаючи міжнародний, державний та місцевий. Міжнародні організації розробляють міжнародні договори та угоди, держави впроваджують національні програми, а місцеві органи влади розробляють місцеві ініціативи з охорони природи[15].

6.2 Забруднення навколишнього середовища

Забруднення навколишнього середовища виникає внаслідок змін властивостей середовища, які стають результатом природних або штучних процесів і призводять до погіршення його функціональних характеристик щодо будь-якого біологічного або технологічного об'єкта. Використовуючи елементи навколишнього середовища, людина впливає на його якість, і ці зміни часто призводять до небезпечного забруднення.

Сучасний світ важко уявити без комп'ютерів, телевізорів та іншої електроніки. Інформаційні та телекомунікаційні технології, включаючи аспекти екології, стали не тільки складовою розвитку, але й концепцією інформаційного суспільства, новим етапом цивілізаційного розвитку та планети.

Хоча інформаційні технології вважаються менш забруднюючими порівняно з іншими видами діяльності, вони все ще не є повністю екологічно безпечними. Наприклад, ефективність інформаційних мереж залежить від кількості користувачів, тобто від кількості комп'ютерів у мережі. Виготовлення одного персонального комп'ютера вимагає значної кількості матеріалів, порівняно з виготовленням автомобіля. Зберігання та використання таких пристроїв призводять до накопичення великої кількості "електронного сміття".

Електронічні пристрої містять небезпечні токсичні речовини, які, потрапляючи в навколишнє середовище, створюють серйозні загрози для життя людей. Наприклад, ртуть, кадмій, свинець та інші речовини, використовувані в електроніці, мають токсичний вплив на навколишнє середовище.

Сучасне сприйняття суспільства щодо проблем екології та суворі екологічні закони змушують виробників створювати системи для збору та утилізації застарілої техніки. Також відбувається збільшення використання переробних матеріалів у конструкції обладнання, що забезпечує більш ефективну утилізацію.

Всі пристрої офісної техніки включають в себе різноманітні органічні та металеві компоненти, придатні для подальшої переробки. З цією метою націлені зусилля на зменшення негативного впливу на природу при виготовленні та використанні офісних пристроїв[15].

6.3 Розробка заходів щодо зменшення забруднення навколишнього середовища

Комп'ютери особистого користування, ноутбуки та інша інформаційна техніка широко використовуються в галузі наукових досліджень, промисловості та у повсякденному домашньому використанні. Проте, із розвитком технологій, виникає питання щодо подальшої долі застарілої техніки, яка може бути

застарілою морально чи вийшла з ладу з різних причин, і яка займає простір в підсобних приміщеннях і складах.

Таблиця 6.1 Шкідливі речовини в ПК

Найменування	Дорогоцінні метали		Кольорові і чорні метали			Полімери і скло	
	Au, г	Ag, г	Al, кг	Cu, кг	Fe, кг	Плас-тик, кг	Скло, кг
Персональний комп'ютер (монітор, системний блок, клавіатура, маніпулятор)	0,053-0,072	0,8-1,1	0,1-0,4	0,1-0,2	3-4	3-3,5	10-20

Процес утилізації офісної та комп'ютерної техніки включає декілька етапів. На початку проводиться списання обладнання безпосередньо на підприємстві. Потім відбувається розбір техніки та сортування матеріалів, при цьому деякі компоненти, які містять цінні метали, наприклад, кінескоп, відправляються на очищення.

Оскільки комп'ютери містять різноманітні метали, такі як золото, срібло, алюміній та інші, покращення утилізації можливе завдяки вторинній переробці.

Основна ідея цього процесу полягає в здійсненні ефективного використання сировини для видобутку корисних і рідкісних матеріалів, таких як іридій та мідь. Цей метод є більш економічно ефективним у порівнянні з традиційним видобутком металів з гірських порід.

Інноваційним підходом до утилізації друкованих плат є використання спеціального розчину, який розчиняється у гарячій воді та дозволяє відшаровувати електронні компоненти.

Це вдосконалення дозволяє повторно використовувати до 90% компонентів нових друкованих плат, що є значним прогресом порівняно з традиційними методами, де це число складає лише 2%.

Багато підприємств стикаються з труднощами утилізації комп'ютерів та обладнання через потребу в спеціалізованому обладнанні та фахових знаннях. Тому тільки досвідчені фахівці можуть успішно впоратися з цією роботою.

Проблема утилізації набуває актуальності, оскільки обсяги виробництва інформаційно-телекомунікаційних технологій та частота їх оновлення змушують компанії звертати увагу на екологічні аспекти. Успішні розробки в даній галузі допоможуть компаніям зменшити податкові витрати на утилізацію застарілих моделей, роблячи екологізацію економічно вигідною[15].

Висновок

Охорона навколишнього середовища є важливою складовою сталого розвитку, спрямованою на збереження та раціональне використання природних ресурсів. Основні завдання охорони включають захист повітря, води та ґрунтів, контроль за використанням природних ресурсів, захист біорізноманіття, зменшення рівня забруднення та відновлення природних ресурсів. Утилізація комп'ютерного обладнання на даний час є актуальною так як все більше підприємств використовує машини в своїй роботі.

ЗАГАЛЬНІ ВИСНОВКИ

1) Проаналізовано особливості процесів з організації управління складським господарством пивзаводу та сучасний стан інформаційного забезпечення; визначено потенційні системи, які можуть бути введені та використовуватися для управління складом.

2) Запропоновано основні принципи розробки інформаційної системи складського господарства пивзаводу, в основу проектування якої покладено задачу комівояжера з складанням технічного та робочого проектів.

3) На мові програмування Delphi розроблено основні проєктні рішення щодо програмного забезпечення, створено фізичну модель бази даних.

4) Розглянуто заходи щодо підвищення економічних показників, розраховано та визначено строк окупності та кошторис на виробництво і впровадження.

5) Проаналізовано фактори ризику, які можуть виникати під час роботи за персональним комп'ютером та розглянуто заходи щодо охорони навколишнього середовища при утилізації комп'ютерного обладнання.

Наукова новизна роботи полягає в наступному:

- вдосконалено підхід до управління процес обробки і систематизації даних про продукцію, сировину, споживачів, документацію, що на відміну від існуючих, дозволяє підвищити ефективність виконання розв'язання логістичних задач.
- вперше для складського господарства пивзаводу запропоновано теоретично-практичне підґрунтя кординації цілей з розширенням можливостей та виходом на міжнародний ринок в умовах воєнного стану.

Наукове значення роботи полягає в отриманні результатів, які будуть спрямовані на вирішення проблем інформаційного забезпечення при управлінні складським господарством на харчовому підприємстві в умовах воєнного стану.

Практичне значення роботи полягає в розробці заходів з інформаційної підтримки підприємства харчової промисловості в умовах наслідків часткової руйнації та дефіциту прісної води у місті Миколаєві.

Публікації за результатами магістерської роботи:

Залітко А.С. Особливості впровадження інформаційних систем на харчових підприємствах України / Залітко А.С. // Сучасні аспекти модернізації науки: стан, проблеми, тенденції розвитку: матеріали XXXIX Міжнародної науково-практичної конференції // м. Гетеборг, Швеція: ГО «ВАДНД», 2023 – С. 447-449.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. "Промислове виробництво пива", підручник для професійно-технічних навчальних закладів, автори: С.В. Бойко, А.І. Бублик, О.П. Гончаренко, В.М. Ковальчук, О.М. Кухаренко, О.В. Кушнір, Л.М. Савчук, В.В. Шевчук, 2018 р.
2. "Складське господарство підприємств", навчальний посібник, автори: В.В. Ситник, О.В. Шинкаренко, 2017 р.
3. "Автоматизовані системи управління складом", навчальний посібник, автори: В.В. Ситник, О.В. Шинкаренко, 2018 р.
4. <https://wezom.com.ua/blog/sap-programmy-cto-eto>
5. <https://www.oracle.com/scm/logistics/warehouse-management/what-is-warehouse-management/>
6. <https://thecrmteam.com/what-are-the-benefits-of-microsoft-dynamics-365/>
7. Грибов, В. А. Складское хозяйство: Учебник. - 2-е изд., перераб. и доп. - М.: ИНФРА-М, 2022. - 320 с.
8. Котляр, В. А. Автоматизация складского хозяйства: Учебное пособие. - М.: ФОРУМ, 2021. - 288 с.
9. Шереметьев, В. В. Учет и контроль запасов: Учебное пособие. - М.: ИНФРА-М, 2022. - 256 с.
10. https://uk.wikipedia.org/wiki/Задача_комівояжера
11. База даних: https://uk.wikipedia.org/wiki/База_даних – Вікіпедія
12. Схема бази даних: https://uk.wikipedia.org/wiki/Схема_бази_даних – Вікіпедія
13. https://vuzlit.com/1006620/analiz_shkidlivih_nebezpechnih_virobnichih_faktoriv_roboti_kompyuteri -5.2/5.3
14. <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/21370/5363.pdf>
5.4.1
15. <https://studfile.net/preview/1852218/page:8/>

Додаток А – код розробки

Код модуля mainUnit.pas

```
unit mainUnit;

interface

uses
    Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
    System.Classes, Vcl.Graphics,
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Menus, Data.DBXMySQL,
    Data.DB, Data.SqlExpr,
    customersUnit, suppliersUnit, materialsUnit, emplUnit, routeUnit,
    invoiceUnit, prodUnit, invPrintUnit,
    Data.Win.ADODB, dm, Vcl.StdCtrls, Vcl.ExtCtrls,
    Vcl.Imaging.pngimage,
    Vcl.Imaging.jpeg;

type
    TmainForm = class(TForm)
        mainConnection: TADOConnection;
        MainMenu1: TMainMenu;
        N1: TMenuItem;
        N2: TMenuItem;
        N3: TMenuItem;
        N4: TMenuItem;
        N5: TMenuItem;
        N6: TMenuItem;
        N7: TMenuItem;
        N8: TMenuItem;
        N11: TMenuItem;
        EditLog: TEdit;
        EditPass: TEdit;
        Label1: TLabel;
        Label2: TLabel;
        CheckBox1: TCheckBox;
        Button1: TButton;
        Logo: TLabel;
        N12: TMenuItem;
        Image1: TImage;
        procedure N2Click(Sender: TObject);
        procedure N3Click(Sender: TObject);
        procedure N4Click(Sender: TObject);
        procedure N5Click(Sender: TObject);
        procedure N6Click(Sender: TObject);
        procedure N7Click(Sender: TObject);
        procedure N8Click(Sender: TObject);
        procedure CheckBox1Click(Sender: TObject);
        procedure Button1Click(Sender: TObject);
        procedure N12Click(Sender: TObject);
        procedure N11Click(Sender: TObject);
    private
        { Private declarations }
    end;

end;
```

```

public
    procedure AddQuery(Query:tadoquery);
    { Public declarations }
end;

var
    MainForm: TmainForm;

implementation

{$R *.dfm}

procedure TmainForm.Button1Click(Sender: TObject);
var
    i:integer;
begin
    if (editLog.Text <> '') or (editPass.Text <> '') then
        showMessage('Логін або пароль невірний')
    else
        begin
            for i := 0 to ControlCount - 1 do
                if (Controls[i] is TButton) or (Controls[i] is TLabel) or
(Controls[i] is Tedit)
                    or (Controls[i] is Tcheckbox) then Controls[i].Visible:=false;
            Logo.Visible:= true;
            Image1.Visible:=true;
            Logo.Caption:= 'Вітаємо в системі';
            mainmenu1.Items[0].Enabled:=true;
            mainmenu1.Items[1].Enabled:=true;
            mainmenu1.Items[2].Enabled:=true;
        end;
end;

procedure TmainForm.CheckBox1Click(Sender: TObject);
begin
    if checkbox1.Checked then editPass.PasswordChar:= #0
    else editPass.PasswordChar:= '*';
end;

procedure TmainForm.N11Click(Sender: TObject);
begin
    MessageBox(handle, PChar('Програма була розроблена для дипломної
роботи'), PChar('Про програму'), MB_OK+MB_ICONINFORMATION);
end;

procedure TmainForm.N12Click(Sender: TObject);
var ques:word;
begin
    ques:=MessageBox(handle, PChar('Ви дійсно хочете вийти'),
PChar('Вихід'), MB_YESNO+MB_ICONQUESTION);
    case ques of
        idyes:Application.Terminate;
    end;
end;

```

```

        end;
    end;

    procedure TmainForm.N2Click(Sender: TObject);
    begin
        invoiceForm.Show;
    end;

    procedure TmainForm.N3Click(Sender: TObject);
    begin
        customersForm.Show;
    end;

    procedure TmainForm.N4Click(Sender: TObject);
    begin
        suppliersForm.Show;
    end;

    procedure TmainForm.N5Click(Sender: TObject);
    begin
        prodForm.Show;
    end;

    procedure TmainForm.N6Click(Sender: TObject);
    begin
        materialsForm.Show;
    end;

    procedure TmainForm.N7Click(Sender: TObject);
    begin
        emplForm.Show;
        emplForm.DBGrid1.Columns[0].Width:= 60;
        emplForm.DBGrid1.Columns[1].Width:= 150;
        emplForm.DBGrid1.Columns[2].Width:= 280;
        emplForm.DBGrid1.Columns[3].Width:= 120;
    end;

    procedure TmainForm.N8Click(Sender: TObject);
    begin
        routeForm.Show;
    end;

    procedure TmainForm.AddQuery(Query:tadoquery);
    begin
        try
            if Query.Modified then Query.Post;
        except
            showMessage('Одне або декілька полів пусті. Запис не може бути
            додано');
        end;
    end;

end.

```


Код модуля customersUnit.pas

```
unit customersUnit;

interface

uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
  System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Data.FMTBcd,
  Data.SqlExpr,
  Vcl.Grids, Vcl.DBGrids, Datasnap.DBClient, Datasnap.Provider,
  Data.Win.ADODB,
  Vcl.ExtCtrls, Vcl.DBCtrls, Vcl.StdCtrls, Vcl.Mask;

type
  TcustomersForm = class(TForm)
    DBGrid1: TDBGrid;
    dsCust: TDataSource;
    ADOQuery1: TADOQuery;
    DBNavigator1: TDBNavigator;
    Panel1: TPanel;
    Label5: TLabel;
    Edit1: TEdit;
    Button1: TButton;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBEdit3: TDBEdit;
    DBEdit4: TDBEdit;
    DBEdit5: TDBEdit;
    DBEdit6: TDBEdit;
    DBEdit7: TDBEdit;
    DBEdit8: TDBEdit;
    DBEdit9: TDBEdit;
    DBEdit10: TDBEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
```

```

    procedure FormShow(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    customersForm: TcustomersForm;

implementation

uses mainUnit, suppliersUnit, materialsUnit, emplUnit, routeUnit,
invoiceUnit, prodUnit, dm;
{$R *.dfm}

procedure TcustomersForm.Button1Click(Sender: TObject);
begin
    Edit1.Clear;
end;

procedure TcustomersForm.Button2Click(Sender: TObject);
begin
    ADOQuery1.Append;
end;

procedure TcustomersForm.Button3Click(Sender: TObject);
begin
    mainform.AddQuery(AdoQuery1)
end;

procedure TcustomersForm.Button4Click(Sender: TObject);
begin
    ADOQuery1.Delete;
end;

procedure TcustomersForm.Edit1Change(Sender: TObject);
var
    srchText, srchText1: string;
begin
    srchText:= '%' + Edit1.Text + '%';
    srchText1:= QuotedStr(srchText);
    adoquery1.close;
    adoquery1.sql.clear;
    adoquery1.sql.add('SELECT * FROM customers WHERE cust_name LIKE '
+ srchText1 +
    'or dir_name LIKE' + srchText1 + 'or addr LIKE' + srchText1 + 'or
ind LIKE' + srchText1
    + 'or phone LIKE' + srchText1 + 'or email LIKE' + srchText1 + 'or
curr_acc LIKE' + srchText1
    + 'or mfo LIKE' + srchText1 + 'or edrpou LIKE' + srchText1);
    adoquery1.open;
    DBGrid1.Columns[0].Width:= 35;

```

```

    DBGrid1.Columns[1].Width:= 180;
    DBGrid1.Columns[2].Width:= 180;
    DBGrid1.Columns[3].Width:= 200;
    DBGrid1.Columns[4].Width:= 50;
    DBGrid1.Columns[5].Width:= 100;
    DBGrid1.Columns[6].Width:= 200;
    DBGrid1.Columns[7].Width:= 250;
    DBGrid1.Columns[8].Width:= 80;
    DBGrid1.Columns[9].Width:= 100;
end;

procedure TcustomersForm.FormShow(Sender: TObject);
begin
    DBGrid1.Columns[0].Width:= 35;
    DBGrid1.Columns[1].Width:= 180;
    DBGrid1.Columns[2].Width:= 180;
    DBGrid1.Columns[3].Width:= 200;
    DBGrid1.Columns[4].Width:= 50;
    DBGrid1.Columns[5].Width:= 100;
    DBGrid1.Columns[6].Width:= 200;
    DBGrid1.Columns[7].Width:= 250;
    DBGrid1.Columns[8].Width:= 80;
    DBGrid1.Columns[9].Width:= 100;
end;

end.

```

Код модуля emplUnit.pas

```

unit emplUnit;

interface

uses
    Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
    System.Classes, Vcl.Graphics,
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Data.FMTBcd,
    Datasnap.DBClient,
    Datasnap.Provider, Data.SqlExpr, Vcl.Grids, Vcl.DBGrids,
    Vcl.StdCtrls, Vcl.Mask, Vcl.ExtCtrls, Vcl.DBCtrls, Data.Win.ADODB;

type
    TemplForm = class(TForm)
        DBGrid1: TDBGrid;
        DBEditEmplId: TDBEdit;
        DBEditEmplPos: TDBEdit;
        DBEditEmplName: TDBEdit;
        DBNavigator1: TDBNavigator;
        Label1: TLabel;
        Label2: TLabel;
        Label3: TLabel;
        DBEditEmplPh: TDBEdit;
        Label4: TLabel;
    end;

```

```

    Panel1: TPanel;
    Label5: TLabel;
    Edit1: TEdit;
    Button1: TButton;
    ADOQuery1: TADOQuery;
    dsEmpl: TDataSource;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    procedure Edit1Change(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    emplForm: TemplForm;

implementation

uses mainUnit, suppliersUnit, materialsUnit, customersUnit,
routeUnit, invoiceUnit, prodUnit, dm;
{$R *.dfm}

procedure TemplForm.Button1Click(Sender: TObject);
begin
    Edit1.Clear;
end;

procedure TemplForm.Button2Click(Sender: TObject);
begin
    ADOQuery1.Append;
end;

procedure TemplForm.Button3Click(Sender: TObject);
begin
    mainform.AddQuery(AdoQuery1)
end;

procedure TemplForm.Button4Click(Sender: TObject);
begin
    ADOQuery1.Delete;
end;

procedure TemplForm.Edit1Change(Sender: TObject);
var
    srchText, srchText1: string;
begin

```

```

    srchText:= '%' + Edit1.Text + '%';
    srchText1:= QuotedStr(srchText);
    adoquery1.close;
    adoquery1.sql.clear;
    adoquery1.sql.add('SELECT * FROM employee WHERE position LIKE ' +
srchText1 +
    'or empl_name LIKE' + srchText1 + 'or phone LIKE' + srchText1);
    adoquery1.open;
    //60 80 250 100
    DBGrid1.Columns[0].Width:= 35;
    DBGrid1.Columns[1].Width:= 180;
    DBGrid1.Columns[2].Width:= 250;
    DBGrid1.Columns[3].Width:= 100;
end;

end.

```

Код модуля invoiceUnit.pas

```

unit invoiceUnit;

interface

uses
    Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
    System.Classes, Vcl.Graphics,
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Data.FMTBcd,
    Datasnap.DBClient,
    Datasnap.Provider, Data.SqlExpr, Vcl.Grids, Vcl.DBGrids,
    Vcl.StdCtrls, Vcl.ExtCtrls, Vcl.DBCtrls, Vcl.Mask, Data.Win.ADODB,
    Vcl.WinXCalendars, Vcl.ComCtrls, Vcl.WinXPickers;

type
    TinvoiceForm = class(TForm)
        DBGrid1: TDBGrid;
        DBNavigator1: TDBNavigator;
        Panel1: TPanel;
        Label5: TLabel;
        Edit1: TEdit;
        Button1: TButton;
        dsProd: TDataSource;
        RadioGroup1: TRadioGroup;
        Button2: TButton;
        Button3: TButton;
        Button4: TButton;
        dsMat: TDataSource;
        DBEdit1: TDBEdit;
        DBLookupComboBox1: TDBLookupComboBox;
        DBLookupComboBox3: TDBLookupComboBox;
        DBEdit3: TDBEdit;
        DBEdit4: TDBEdit;
        DBEdit5: TDBEdit;
        DBLookupComboBox4: TDBLookupComboBox;
    end;

```

```

    DBLookupComboBox5: TDBLookupComboBox;
    DBLookupComboBox6: TDBLookupComboBox;
    DBEdit6: TDBEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    DBLookupComboBox7: TDBLookupComboBox;
    ComboBox1: TComboBox;
    DBLookupComboBox2: TDBLookupComboBox;
    Button5: TButton;
    Button6: TButton;
    EditStartdate: TEdit;
    Editenddate: TEdit;
    Button7: TButton;
    Label11: TLabel;
    Label12: TLabel;
    procedure RadioGroup1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure Button6Click(Sender: TObject);
    procedure Button7Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    invoiceForm: TinvoiceForm;
    date: tdatetime;

implementation

uses mainUnit, suppliersUnit, materialsUnit, emplUnit, routeUnit,
customersUnit, prodUnit,
dm, invPrintUnit, repProdUnit, repMatUnit;

{$R *.dfm}

procedure TinvoiceForm.Button3Click(Sender: TObject);
begin
    case Radiogroup1.ItemIndex of
    0: begin

```

```

        mainForm.AddQuery(dm.DataModule1.ADOQueryProd)
    end;
    1: begin
        mainForm.AddQuery(dm.DataModule1.ADOQueryMat)
    end;
    end;
end;

procedure TinvoiceForm.Button5Click(Sender: TObject);
begin
    printInvForm.Show;
    if radiogroup1.ItemIndex = 0 then
        printInvForm.frxReport1.ShowReport()
    else
        printinvform.frxReport2.ShowReport();
end;

//date search
procedure TinvoiceForm.Button6Click(Sender: TObject);
begin
    case radiogroup1.ItemIndex of
    0: begin
        dm.DataModule1.ADOQueryProd.close;
        dm.DataModule1.ADOQueryProd.sql.clear;
        dm.DataModule1.ADOQueryProd.sql.add
            ('SELECT * FROM invoice ');
        dm.DataModule1.ADOQueryProd.sql.add
            ('WHERE route_id = 1 and trans_date between :RangeStart and
:RangeEnd');

        dm.DataModule1.ADOQueryProd.Parameters.ParamByName('RangeStart').Value
        := editstartdate.Text;

        dm.DataModule1.ADOQueryProd.Parameters.ParamByName('RangeEnd').Value
        := editenddate.Text;
        dm.DataModule1.ADOQueryProd.open;
    end;
    1: begin
        dm.DataModule1.ADOQuerymat.close;
        dm.DataModule1.ADOQuerymat.sql.clear;
        dm.DataModule1.ADOQuerymat.sql.add
            ('SELECT * FROM invoice ');
        dm.DataModule1.ADOQuerymat.sql.add
            ('WHERE route_id = 2 and trans_date between :RangeStart and
:RangeEnd');

        dm.DataModule1.ADOQuerymat.Parameters.ParamByName('RangeStart').Value
        := editstartdate.Text;

        dm.DataModule1.ADOQuerymat.Parameters.ParamByName('RangeEnd').Value
        := editenddate.Text;
        dm.DataModule1.ADOQuerymat.open;
    end;
end;

```

```

    end;
end;

procedure TinvoiceForm.Button7Click(Sender: TObject);
begin
    if radiogroup1.ItemIndex = 0 then
    begin
        repprodunit.repProdFrom.Show;
        repprodunit.repProdFrom.frxReport1.ShowReport();
    end else
        repMatForm.Show;
        repMatForm.frxReport1.ShowReport();
end;

procedure TinvoiceForm.Edit1Change(Sender: TObject);
var
    srchText, srchText1: string;
begin
    case RadioGroup1.ItemIndex of
        //radigroup products
        0: begin
            srchText:= '%' + Edit1.Text + '%';
            srchText1:= QuotedStr(srchText);
            dm.DataModule1.ADOQueryProd.close;
            dm.DataModule1.ADOQueryProd.sql.clear;
            case combobox1.ItemIndex of
                //співроб
                0: begin
                    dm.DataModule1.ADOQueryProd.sql.add
                        ('SELECT * FROM invoice WHERE route_id = 1 and empl_id IN
(select empl_id from employee where empl_name like' +srchText1+')');
                    end;
                    //замовник/постачальник
                1: begin
                    dm.DataModule1.ADOQueryProd.sql.add
                        ('SELECT * FROM invoice WHERE cust_id IN (select cust_id from
customers where cust_name like' +srchText1+')');
                    end;
                    //песуц/товар
                2: begin
                    dm.DataModule1.ADOQueryProd.sql.add
                        ('SELECT * FROM invoice WHERE prod_id IN (select prod_id from
products where prod_name like' +srchText1+')');
                    end;
                end;
            dm.DataModule1.ADOQueryProd.open;
            end;
        //radigroup materials
        1: begin
            srchText:= '%' + Edit1.Text + '%';
            srchText1:= QuotedStr(srchText);
            dm.DataModule1.ADOQuerymat.close;
            dm.DataModule1.ADOQuerymat.sql.clear;

```



```

case combobox1.ItemIndex of
//співроб
0: begin
    dm.DataModule1.ADOQuerymat.sql.add
        ('SELECT * FROM invoice WHERE route_id = 2 and empl_id IN
(select empl_id from employee where empl_name like' +srchText1+')');
    end;
    //замовник/постачальник
1: begin
    dm.DataModule1.ADOQuerymat.sql.add
        ('SELECT * FROM invoice WHERE cust_id IN (select cust_id from
suppliers where cust_name like' +srchText1+')');
    end;
    //песує/товар
2: begin
    dm.DataModule1.ADOQuerymat.sql.add
        ('SELECT * FROM invoice WHERE prod_id IN (select prod_id from
materials where prod_name like' +srchText1+')');
    end;
end;
dm.DataModule1.ADOQuerymat.open;
end;
end;
//setting previous table
if edit1.Text = '' then
if RadioGroup1.ItemIndex = 0 then
begin
    dm.DataModule1.ADOQueryProd.close;
    dm.DataModule1.ADOQueryProd.sql.clear;
    dm.DataModule1.ADOQueryProd.sql.add('SELECT inv_id, empl_id,
route_id, cust_id, prod_id, meas_unit, weight, inv_creation_date,
trans_date FROM invoice WHERE route_id = 1');
    dm.DataModule1.ADOQueryProd.open;
end
else
begin
    dm.DataModule1.ADOQueryMat.close;
    dm.DataModule1.ADOQueryMat.sql.clear;
    dm.DataModule1.ADOQueryMat.sql.add('SELECT inv_id, empl_id,
route_id, cust_id, suppl_id, material_id, meas_unit, weight,
inv_creation_date, trans_date FROM invoice WHERE route_id = 2');
    dm.DataModule1.ADOQueryMat.open;
end;
end;
//disabling all
procedure TinvoiceForm.FormCreate(Sender: TObject);
var
    i:integer;
begin
    for i := 0 to invoiceForm.ControlCount - 1 do
        if Controls[i] is TRadioGroup then Controls[i].Enabled:=true
        else
            Controls[i].Enabled:=false;

```

```

end;

procedure TinvoiceForm.Button1Click(Sender: TObject);
begin
    Edit1.Clear;
end;

//append
procedure TinvoiceForm.Button2Click(Sender: TObject);
begin
    case Radiogroup1.ItemIndex of
        0: begin
            dm.DataModule1.ADOQueryProd.Append;
        end;
        1: begin
            dm.DataModule1.ADOQueryMat.Append;
        end;
    end;
end;

//changing work mode
procedure changing(var a, b, c, d, e:tdbedit; grid:tdbgrid;
ds:tdatasource;
dbl1, dbl2, dbl3, dbl4, dbl5, dbl6, dbl7:tdblookupcombobox);
begin
    grid.DataSource:= ds;
    a.DataSource:= ds;
    b.DataSource:= ds;
    c.DataSource:= ds;
    d.DataSource:= ds;
    e.DataSource:= ds;
    dbl1.Visible:= True;
    dbl2.Visible:= True;
    dbl3.Visible:= True;
    dbl4.Visible:= False;
    dbl5.Visible:= False;
    dbl6.Visible:= False;
    dbl7.DataSource:= ds;
end;

procedure TinvoiceForm.RadioGroup1Click(Sender: TObject);
var i:integer;
begin
    for i := 0 to invoiceForm.ControlCount - 1 do
        Controls[i].Enabled:=true;
    case RadioGroup1.ItemIndex of
        0: begin
            //вибір всього в ґрид
            dm.DataModule1.ADOQueryProd.close;
            dm.DataModule1.ADOQueryProd.sql.clear;
            dm.DataModule1.ADOQueryProd.sql.add('SELECT inv_id, empl_id,
route_id, cust_id, prod_id, meas_unit, weight, inv_creation_date,
trans_date FROM invoice WHERE route_id = 1');

```

```

dm.DataModule1.ADOQueryProd.open;
label4.Caption:= 'Замовник';
label6.Caption:= 'Продукція';
changing(dbedit1, dbedit3, dbedit4, dbedit5, dbedit6, dbgrid1,
dsProd, dblookupcombobox1, dblookupcombobox3,
dblookupcombobox5,
dblookupcombobox2, dblookupcombobox4, dblookupcombobox6,
dblookupcombobox7)
end;
1: begin
dm.DataModule1.ADOQueryMat.close;
dm.DataModule1.ADOQueryMat.sql.clear;
dm.DataModule1.ADOQueryMat.sql.add('SELECT inv_id, empl_id,
route_id, cust_id, suppl_id, material_id, meas_unit, weight,
inv_creation_date, trans_date FROM invoice WHERE route_id = 2');
dm.DataModule1.ADOQueryMat.open;
label4.Caption:= 'Постачальник';
label6.Caption:= 'Матеріал';
changing(dbedit1, dbedit3, dbedit4, dbedit5, dbedit6, dbgrid1,
dsMat, dblookupcombobox2, dblookupcombobox4,
dblookupcombobox6, dblookupcombobox1,
dblookupcombobox3, dblookupcombobox5, dblookupcombobox7)
end;
end;
end;
end.

```

Код модуля materialsUnit.pas

```

unit materialsUnit;

interface

uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
  System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Data.FMTBcd,
  Datasnap.DBClient,
  Datasnap.Provider, Data.SqlExpr, Vcl.Grids, Vcl.DBGrids, Vcl.Mask,
  Vcl.ExtCtrls, Vcl.DBCtrls, Data.Win.ADODB, Vcl.StdCtrls;

type
  TmaterialsForm = class(TForm)
    DBGrid1: TDBGrid;
    Panel1: TPanel;
    Label5: TLabel;
    Edit1: TEdit;
    Button1: TButton;
    DBNavigator1: TDBNavigator;
    ADOQuery1: TADOQuery;
    dsMater: TDataSource;
  end;

```

```

    Label1: TLabel;
    Ціна: TLabel;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBEdit3: TDBEdit;
    Label2: TLabel;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    DBEdit4: TDBEdit;
    DBEdit5: TDBEdit;
    Label3: TLabel;
    Label4: TLabel;
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
    procedure FormShow(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    materialsForm: TmaterialsForm;

implementation

uses mainUnit, suppliersUnit, customersUnit, emplUnit, routeUnit,
invoiceUnit, prodUnit, dm;
{$R *.dfm}

procedure TmaterialsForm.Button1Click(Sender: TObject);
begin
    Edit1.Clear;
end;

procedure TmaterialsForm.Button2Click(Sender: TObject);
begin
    ADOQuery1.Append;
end;

procedure TmaterialsForm.Button3Click(Sender: TObject);
begin
    mainform.AddQuery(AdoQuery1)
end;

procedure TmaterialsForm.Button4Click(Sender: TObject);
begin
    ADOQuery1.Delete;
end;

```

```

procedure TmaterialsForm.Edit1Change(Sender: TObject);
var
    srchText, srchText1: string;
begin
    srchText:= '%' + Edit1.Text + '%';
    srchText1:= QuotedStr(srchText);
    adoquery1.close;
    adoquery1.sql.clear;
    adoquery1.sql.add('SELECT * FROM materials WHERE material_name
LIKE ' + srchText1 +
    'or price LIKE' + srchText1);
    adoquery1.open;
    DBGrid1.Columns[0].Width:= 35;
    DBGrid1.Columns[1].Width:= 200;
    DBGrid1.Columns[2].Width:= 35;
    DBGrid1.Columns[3].Width:= 60;
    DBGrid1.Columns[4].Width:= 70;
end;

procedure TmaterialsForm.FormShow(Sender: TObject);
begin
    DBGrid1.Columns[0].Width:= 35;
    DBGrid1.Columns[1].Width:= 200;
    DBGrid1.Columns[2].Width:= 35;
    DBGrid1.Columns[3].Width:= 60;
    DBGrid1.Columns[4].Width:= 70;
end;

end.

```

Код модуля prodUnit.pas

```

unit prodUnit;

interface

uses
    Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
    System.Classes, Vcl.Graphics,
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Data.FMTBcd,
    Datasnap.DBClient,
    Datasnap.Provider, Data.SqlExpr, Vcl.Grids, Vcl.DBGrids,
    Data.Win.ADODB,
    Vcl.ExtCtrls, Vcl.DBCtrls, Vcl.StdCtrls, Vcl.Mask;

type
    TprodForm = class(TForm)
        DBGrid1: TDBGrid;
        DBNavigator1: TDBNavigator;
        ADOQuery1: TADOQuery;
        dsProd: TDataSource;
        Panell: TPanel;
    end;

```

```

    Label5: TLabel;
    Edit1: TEdit;
    Button1: TButton;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBEdit3: TDBEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    DBEdit4: TDBEdit;
    Label4: TLabel;
    DBEdit5: TDBEdit;
    Label6: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    prodForm: TprodForm;

implementation

uses mainUnit, suppliersUnit, materialsUnit, emplUnit, routeUnit,
invoiceUnit, customersUnit, dm;

{$R *.dfm}

procedure TprodForm.Button1Click(Sender: TObject);
begin
    Edit1.Clear;
end;

procedure TprodForm.Button2Click(Sender: TObject);
begin
    ADOQuery1.Append;
end;

procedure TprodForm.Button3Click(Sender: TObject);
begin
    mainform.AddQuery(AdoQuery1)
end;

```

```

procedure TprodForm.Button4Click(Sender: TObject);
begin
    ADOQuery1.Delete
end;

procedure TprodForm.Edit1Change(Sender: TObject);
var
    srchText, srchText1: string;
begin
    srchText:= '%' + Edit1.Text + '%';
    srchText1:= QuotedStr(srchText);
    adoquery1.close;
    adoquery1.sql.clear;
    adoquery1.sql.add('SELECT * FROM products WHERE prod_type LIKE ' +
srchText1 +
    'or prod_name LIKE' + srchText1 + 'or price LIKE' + srchText1);
    adoquery1.open;
    DBGrid1.Columns[0].Width:= 35;
    DBGrid1.Columns[1].Width:= 120;
    DBGrid1.Columns[2].Width:= 200;
    DBGrid1.Columns[3].Width:= 50;
    DBGrid1.Columns[4].Width:= 60;
    DBGrid1.Columns[5].Width:= 70;
end;

procedure TprodForm.FormShow(Sender: TObject);
begin
    DBGrid1.Columns[0].Width:= 35;
    DBGrid1.Columns[1].Width:= 120;
    DBGrid1.Columns[2].Width:= 200;
    DBGrid1.Columns[3].Width:= 50;
    DBGrid1.Columns[4].Width:= 60;
    DBGrid1.Columns[5].Width:= 70;
end;

procedure AddQuery(Query:tadoquery);
begin
    try
        if Query.Modified then Query.Post;
    except
        showMessage('pizda');
    end;
end;

end.

```

Код модуля suppliersUnit.pas

```

unit suppliersUnit;

interface

uses

```

```

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
System.Classes, Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Data.FMTBcd,
Datanap.DBClient,
Datanap.Provider, Data.SqlExpr, Vcl.Grids, Vcl.DBGrids,
Vcl.ExtCtrls,
Vcl.DBCtrls, Data.Win.ADODB, Vcl.StdCtrls, Vcl.Mask;

```

```

type

```

```

TsuppliersForm = class(TForm)
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    Panel1: TPanel;
    Label5: TLabel;
    Edit1: TEdit;
    Button1: TButton;
    dsSuppl: TDataSource;
    ADOQuery1: TADOQuery;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBEdit3: TDBEdit;
    DBEdit4: TDBEdit;
    DBEdit5: TDBEdit;
    DBEdit6: TDBEdit;
    DBEdit7: TDBEdit;
    DBEdit8: TDBEdit;
    DBEdit9: TDBEdit;
    DBEdit10: TDBEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
    procedure FormShow(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

```



```

var
    suppliersForm: TsuppliersForm;

implementation

uses mainUnit, customersUnit, materialsUnit, emplUnit, routeUnit,
invoiceUnit, prodUnit, dm;
{$R *.dfm}

procedure TsuppliersForm.Button1Click(Sender: TObject);
begin
    Edit1.Clear;
end;

procedure TsuppliersForm.Button2Click(Sender: TObject);
begin
    ADOQuery1.Append;
end;

procedure TsuppliersForm.Button3Click(Sender: TObject);
begin
    mainform.AddQuery(AdoQuery1)
end;

procedure TsuppliersForm.Button4Click(Sender: TObject);
begin
    ADOQuery1.Delete;
end;

procedure TsuppliersForm.Edit1Change(Sender: TObject);
var
    srchText, srchText1: string;
begin
    srchText:= '%' + Edit1.Text + '%';
    srchText1:= QuotedStr(srchText);
    adoquery1.close;
    adoquery1.sql.clear;
    adoquery1.sql.add('SELECT * FROM suppliers WHERE suppl_name LIKE '
+ srchText1 +
    'or dir_name LIKE' + srchText1 + 'or addr LIKE' + srchText1 + 'or
ind LIKE' + srchText1
    + 'or phone LIKE' + srchText1 + 'or email LIKE' + srchText1 + 'or
curr_acc LIKE' + srchText1
    + 'or mfo LIKE' + srchText1 + 'or edrpou LIKE' + srchText1);
    adoquery1.open;
    DBGrid1.Columns[0].Width:= 35;
    DBGrid1.Columns[1].Width:= 180;
    DBGrid1.Columns[2].Width:= 180;
    DBGrid1.Columns[3].Width:= 200;
    DBGrid1.Columns[4].Width:= 50;
    DBGrid1.Columns[5].Width:= 100;
    DBGrid1.Columns[6].Width:= 200;
    DBGrid1.Columns[7].Width:= 250;

```

```

    DBGrid1.Columns[8].Width:= 80;
    DBGrid1.Columns[9].Width:= 100;
end;

procedure TsuppliersForm.FormShow(Sender: TObject);
begin
    DBGrid1.Columns[0].Width:= 35;
    DBGrid1.Columns[1].Width:= 180;
    DBGrid1.Columns[2].Width:= 180;
    DBGrid1.Columns[3].Width:= 200;
    DBGrid1.Columns[4].Width:= 50;
    DBGrid1.Columns[5].Width:= 100;
    DBGrid1.Columns[6].Width:= 200;
    DBGrid1.Columns[7].Width:= 250;
    DBGrid1.Columns[8].Width:= 80;
    DBGrid1.Columns[9].Width:= 100;
end;

end.

```

Код модуля routeUnit.pas

```

unit routeUnit;

interface

uses
    Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
    System.Classes, Vcl.Graphics,
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Data.Win.ADODB,
    Vcl.Grids,
    Vcl.DBGrids;

type
    TrouteForm = class(TForm)
        DBGrid1: TDBGrid;
        ADOQuery1: TADOQuery;
        DataSource1: TDataSource;
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    routeForm: TrouteForm;

implementation

uses mainUnit, suppliersUnit, customersUnit, emplUnit,
materialsUnit, invoiceUnit, prodUnit, dm;
{$R *.dfm}

```

```

end.
Код модуля dmUnit.pas
unit dm;

interface

uses
  System.SysUtils, System.Classes, Data.DB, Data.Win.ADODB;

type
  TDataModule1 = class(TDataModule)
    ADOQueryProd: TADOQuery;
    ADOQueryMat: TADOQuery;
    ADOQueryMatinv_id: TAutoIncField;
    ADOQueryMatempl_id: TIntegerField;
    ADOQueryMatroute_id: TIntegerField;
    ADOQueryMatsuppl_id: TIntegerField;
    ADOQueryMatmaterial_id: TIntegerField;
    ADOQueryMatmeas_unit: TWideStringField;
    ADOQueryMatweight: TWideStringField;
    ADOQueryMatinv_creation_date: TDateField;
    ADOQueryMattrans_date: TDateField;
    ADOQueryProdinv_id: TAutoIncField;
    ADOQueryProdempl_id: TIntegerField;
    ADOQueryProdroute_id: TIntegerField;
    ADOQueryProdcust_id: TIntegerField;
    ADOQueryProdprod_id: TIntegerField;
    ADOQueryProdmeas_unit: TWideStringField;
    ADOQueryProdweight: TWideStringField;
    ADOQueryProdinv_creation_date: TDateField;
    ADOQueryProdtrans_date: TDateField;
    ADOQueryProdempl_id_name: TStringField;
    ADOQueryProdroute_id_name: TStringField;
    ADOQueryProdcust_id_name: TStringField;
    ADOQueryProdprod_id_name: TStringField;
    ADOQueryMatroute_id_name: TStringField;
    ADOQueryMatsuppl_id_name: TStringField;
    ADOQueryMatmaterial_id_name: TStringField;
    ADOQueryMatempl_id_name: TStringField;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  DataModule1: TDataModule1;

implementation

uses suppliersUnit, customersUnit, emplUnit, materialsUnit,
  invoiceUnit, prodUnit, routeUnit, mainUnit;

```

```
{%CLASSGROUP 'Vcl.Controls.TControl'}
```

```
{$R *.dfm}
```

```
end.
```

Код модуля invPrintUnit.pas

```
unit invPrintUnit;
```

```
interface
```

```
uses
```

```
Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,  
System.Classes, Vcl.Graphics,
```

```
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, frxClass, frxPreview,  
frxDBSet,
```

```
Vcl.StdCtrls, Vcl.Menus, frxExportPDF, frxExportBaseDialog,  
frxExportDOCX;
```

```
type
```

```
TprintInvForm = class(TForm)
```

```
frxDBDataset1: TfrxDBDataset;
```

```
frxReport1: TfrxReport;
```

```
frxPreview1: TfrxPreview;
```

```
MainMenu1: TMainMenu;
```

```
N1: TMenuItem;
```

```
N2: TMenuItem;
```

```
PDF1: TMenuItem;
```

```
PDF2: TMenuItem;
```

```
frxDOCXExport1: TfrxDOCXExport;
```

```
frxPDFExport1: TfrxPDFExport;
```

```
frxDBDataset2: TfrxDBDataset;
```

```
frxReport2: TfrxReport;
```

```
procedure N2Click(Sender: TObject);
```

```
procedure PDF1Click(Sender: TObject);
```

```
procedure PDF2Click(Sender: TObject);
```

```
private
```

```
{ Private declarations }
```

```
public
```

```
{ Public declarations }
```

```
end;
```

```
var
```

```
printInvForm: TprintInvForm;
```

```
implementation
```

```
uses suppliersUnit, customersUnit, emplUnit, materialsUnit,  
invoiceUnit, prodUnit, routeUnit, mainUnit;
```

```
{$R *.dfm}
```

```

procedure TprintInvForm.N2Click(Sender: TObject);
begin
    frxreport1.Print;
end;

procedure TprintInvForm.PDF1Click(Sender: TObject);
begin
    frxreport1.Export(printinvform.frxPDFExport1);
end;

procedure TprintInvForm.PDF2Click(Sender: TObject);
begin
    frxreport1.Export(printinvform.frxDOCXExport1);
end;

end.

```

Код модуля repProdUnit.pas

```

unit repProdUnit;

interface

uses
    Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
    System.Classes, Vcl.Graphics,
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, frxClass, frxDBSet,
    frxPreview,
    frxExportPDF, Vcl.Menus, frxExportBaseDialog, frxExportDOCX;

type
    TrepProdForm = class(TForm)
        frxReport1: TfrxReport;
        frxDBDataset1: TfrxDBDataset;
        repProdForm: TfrxPreview;
        frxPreview1: TfrxPreview;
        frxDOCXExport1: TfrxDOCXExport;
        MainMenu1: TMainMenu;
        N1: TMenuItem;
        PDF1: TMenuItem;
        PDF2: TMenuItem;
        N2: TMenuItem;
        frxPDFExport1: TfrxPDFExport;
        procedure PDF1Click(Sender: TObject);
        procedure PDF2Click(Sender: TObject);
        procedure N2Click(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var

```

```

    repProdFrom: TrepProdFrom;

implementation

uses mainUnit, suppliersUnit, materialsUnit, emplUnit, routeUnit,
customersUnit, prodUnit,
invPrintUnit, invoiceUnit;

{$R *.dfm}

procedure TrepProdFrom.N2Click(Sender: TObject);
begin
    frxreport1.Print;
end;

procedure TrepProdFrom.PDF1Click(Sender: TObject);
begin
    frxreport1.Export(printinvform.frxPDFExport1);
end;

procedure TrepProdFrom.PDF2Click(Sender: TObject);
begin
    frxreport1.Export(printinvform.frxDOCXExport1);
end;

end.

```

Код модуля repMatUnit.pas

```

unit repMatUnit;

interface

uses
    Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
    System.Classes, Vcl.Graphics,
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, frxClass, frxPreview,
    frxDBSet,
    frxExportPDF, Vcl.Menus, frxExportBaseDialog, frxExportDOCX;

type
    TrepMatForm = class(TForm)
        frxReport1: TfrxReport;
        frxDBDataset1: TfrxDBDataset;
        frxPreview1: TfrxPreview;
        frxPreview2: TfrxPreview;
        frxDOCXExport1: TfrxDOCXExport;
        MainMenu1: TMainMenu;
        N1: TMenuItem;
        PDF1: TMenuItem;
        PDF2: TMenuItem;
        N2: TMenuItem;
        frxPDFExport1: TfrxPDFExport;
    end;

```

```

        procedure PDF1Click(Sender: TObject);
        procedure PDF2Click(Sender: TObject);
        procedure N2Click(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    repMatForm: TrepMatForm;

implementation

{$R *.dfm}

uses customersUnit, dm, emplUnit, invoiceUnit, invPrintUnit,
    mainUnit,
        materialsUnit, prodUnit, repProdUnit, routeUnit, suppliersUnit;

procedure TrepMatForm.N2Click(Sender: TObject);
begin
    frxreport1.Print;
end;

procedure TrepMatForm.PDF1Click(Sender: TObject);
begin
    frxreport1.Export(printinvform.frxPDFExport1);
end;

procedure TrepMatForm.PDF2Click(Sender: TObject);
begin
    frxreport1.Export(printinvform.frxDOCXExport1);
end;

end.

```

SQL-код створення бази даних

```

# Створити базу даних
CREATE DATABASE db_first;

# Увійти в базу даних
USE db_first;

# Створити таблицю Employee
CREATE TABLE Employee (
    empl_id INT AUTO_INCREMENT PRIMARY KEY,
    position VARCHAR(255) not null,
    empl_name VARCHAR(255) not null,
    phone VARCHAR(255) not null
);

```

```

# Створити таблицю Route_sheet
CREATE TABLE Route_sheet (
    route_id INT AUTO_INCREMENT PRIMARY KEY,
    route_name VARCHAR(255) not null
);

# Створити таблицю Materials
CREATE TABLE Materials (
    material_id INT AUTO_INCREMENT PRIMARY KEY,
    material_name VARCHAR(255) not null,
    price DECIMAL(10,2) not null
);

# Створити таблицю Products
CREATE TABLE Products (
    prod_id INT AUTO_INCREMENT PRIMARY KEY,
    prod_type VARCHAR(255) not null,
    prod_name VARCHAR(255) not null,
    price DECIMAL(10,2) not null
);

# Створити таблицю Customers
CREATE TABLE Customers (
    cust_id INT AUTO_INCREMENT PRIMARY KEY,
    cust_name VARCHAR(255) not null,
    dir_name VARCHAR(255) not null,
    addr VARCHAR(255) not null,
    ind VARCHAR(255) not null,
    phone VARCHAR(255) not null,
    email VARCHAR(255) not null,
    curr_acc VARCHAR(255) not null,
    MFO VARCHAR(255) not null,
    edrpou VARCHAR(255) not null
);

# Створити таблицю Suppliers
CREATE TABLE Suppliers (
    suppl_id INT AUTO_INCREMENT PRIMARY KEY,
    suppl_name VARCHAR(255) not null,
    dir_name VARCHAR(255) not null,
    addr VARCHAR(255) not null,
    ind VARCHAR(255) not null,
    phone VARCHAR(255) not null,
    email VARCHAR(255) not null,
    curr_acc VARCHAR(255) not null,
    MFO VARCHAR(255) not null,
    edrpou VARCHAR(255) not null
);

# Створити таблицю Invoice
CREATE TABLE Invoice (
    inv_id INT AUTO_INCREMENT PRIMARY KEY,
    empl_id INT not null,

```



```
route_id INT not null,  
cust_id INT null,  
prod_id INT null,  
material_id INT null,  
suppl_id INT null,  
meas_unit VARCHAR(255) not null,  
weight VARCHAR(255) not null,  
inv_creation_date DATE not null,  
trans_date DATE not null,  
FOREIGN KEY (empl_id) REFERENCES Employee (empl_id),  
FOREIGN KEY (route_id) REFERENCES Route_sheet (route_id),  
FOREIGN KEY (cust_id) REFERENCES Customers (cust_id),  
FOREIGN KEY (prod_id) REFERENCES Products (prod_id),  
FOREIGN KEY (material_id) REFERENCES Materials (material_id),  
FOREIGN KEY (suppl_id) REFERENCES Suppliers (suppl_id)  
);
```

Додаток Б – тези

Залітко А.С.
магістр,
*Національний університет
кораблебудування
імені адмірала Макарова
м. Миколаїв, Україна*

ОСОБЛИВОСТІ ВПРОВАДЖЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ НА ХАРЧОВИХ ПІДПРИЄМСТВАХ УКРАЇНИ

Спеціалізовані інформаційні системи (ІС) відіграють важливу роль у діяльності підприємств харчової промисловості, забезпечуючи ефективне управління, підвищуючи продуктивність праці та покращуючи якість готової продукції та послуг. Це складний процес, який вимагає ретельної підготовки та планування. Умови воєнного стану в Україні додають додаткових викликів, які пов'язані зі:

- зміною пріоритетів: у воєнний час підприємствам необхідно зосередитися на завданнях, що забезпечать їх стабільність та виживання. У таких умовах впровадження ІС може відкладатися або змінювати пріоритети їхнього впровадження. Наприклад, підприємства харчової промисловості можуть зосередитися на впровадженні ІС, які допомагають краще адаптуватися до нових умов або забезпечують безперебійну роботу критично важливих систем;
- збільшенням ризиків: впровадження ІС на підприємстві має на увазі певні ризики, які в умовах воєнного стану очевидно можуть зростати. Зростає ризик втрати даних, кібератак або пошкодження обладнання. Підприємствам необхідно враховувати ці ризики та приймати певні заходи для їх мінімізації;
- нестачею ресурсів: в умовах воєнного стану підприємства харчової промисловості можуть мати обмежені ресурси, які необхідні для впровадження ІС. Наприклад, обмежені фінанси, нестача персоналу або обладнання. Планувати

впровадження ІС підприємствам необхідно більш ретельно та з урахуванням наявних ресурсів [1].

Далі наведені такі рекомендації для успішного впровадження ІС в умовах воєнного стану на підприємствах харчової промисловості:

- ретельний план впровадження. План впровадження необхідний при створенні будь-якої ІС, особливо ретельно слід розробити його в умовах воєнного стану. Він повинен включати наступні елементи: цілі впровадження, перелік ресурсів, етапи реалізації, терміни розробки та фінансування;
- зосередження на найважливіших завданнях. Підприємствам харчової промисловості необхідно визначити найважливіші завдання, які повинна вирішувати ІС. Якщо дані завдання не є критичними для підприємства в даний час, їх можна відкласти;
- враховування ризиків. Необхідно провести оцінку ризиків, які пов'язані з впровадженням ІС, та розробити заходи їх мінімізації. Зокрема, необхідно забезпечити безпеку ІС, захист від кібератак, а також резервне копіювання даних, для їх збереження;
- використання хмарних рішень. Є актуальним в умовах воєнного стану. Хмарні сервіси надаються віддалено, в режимі онлайн. Це дозволяє підприємствам отримувати доступ до даних з будь-якого місця. Дані рішення можуть бути особливо важливими для підприємств, які повністю або частково були змушені переїхати в інші регіони країни [2].

Таким чином, розглянуто особливості впровадження інформаційних систем на підприємствах в умовах воєнного стану. Зосереджено увагу на зміні пріоритетів, збільшенні ризиків та нестачі ресурсів. Надано практичні рекомендації щодо ретельного планування впровадження спеціалізованих інформаційних систем та зосередження на найважливіших завданнях з урахуванням ризиків і використанням хмарних технологій.

Список використаних джерел:

[1] Безус А.М. Особливості впровадження інформаційних технологій в сучасних умовах // Ефективна економіка. - 2022. - № 4. - С. 58-63.

[2] Когут О.В., Когут Л.М., Шевчук А.В. Особливості впровадження інформаційних систем на підприємствах в умовах війни // Вісник Кременчуцького національного університету імені Михайла Остроградського. Серія: Економічні науки. - 2023. - № 1. - С. 55-60.

Додаток В – Презентаційні матеріали

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук і управління проектами
Кафедра інформаційних управляючих систем і технологій

Спеціальність 122 – Комп'ютерні науки

Дипломна робота на тему
«Дослідження технологій та розробка
інформаційної системи для складського
господарства пивзаводу»

Магістерська дипломна робота

Виконавець: студент групи 6142м Залітко А. С.

Керівник дтн. проф. Казимиренко Ю.О.

Миколаїв
2023

Слайд 1

Актуальність роботи



Складське господарство пивзаводу – це система складів, що забезпечує зберігання сировини, матеріалів, готової продукції та інших матеріальних цінностей, необхідних для виробництва пива. Воно є важливою частиною виробництва пива. Правильна організація складського господарства забезпечує ефективне зберігання матеріальних цінностей та мінімізує їхні втрати.

Актуальність дослідження обумовлена кількома факторами, зокрема ростом важливості складської діяльності в логістичних системах, зростанням вимог до ефективності цих процесів та розвитком інформаційних технологій для автоматизації складських процесів.

В умовах відродження виробництва під час військового стану, запуск діяльності пивзаводу є важливою складовою відновлення міста Миколаєва.

Мета, завдання, об'єкт, предмет дослідження дипломного проекту

Метою роботи є дослідження технологій та розробка інформаційної системи для управління процесами складського господарства пивзаводу, впровадження якої спрямовано на систематизацію обліку продукції, ресурсів та скорочення часу на обробку інформації.

Об'єктом дослідження є процес обробки і систематизації даних про продукцію, сировину, споживачів, документацію. Методи дослідження включають аналіз, моделювання та економічні розрахунки.

Завданням же є аналіз сучасного стану інформаційного забезпечення складського господарства пивзаводу, дослідження принципів проектування інформаційних систем та технології баз даних складського господарства, розробка програмного забезпечення для інформаційної системи та економічне обґрунтування

Порівняльна таблиця існуючих рішень

	ORACLE	SAP	MS Dynamics
Широкий функціонал	Так	Так	Ні
Інтеграція з іншими системами	Так	Так	Ні
Простота використання	Ні	Ні	Так
Ведення більш ніж одного підприємства	Так	Так	Так
Складність оновлення	Так	Так	Так
Високі вимоги до апаратної частини	Так	Так	Так

Теоретичні передумови моделювання інформаційної системи

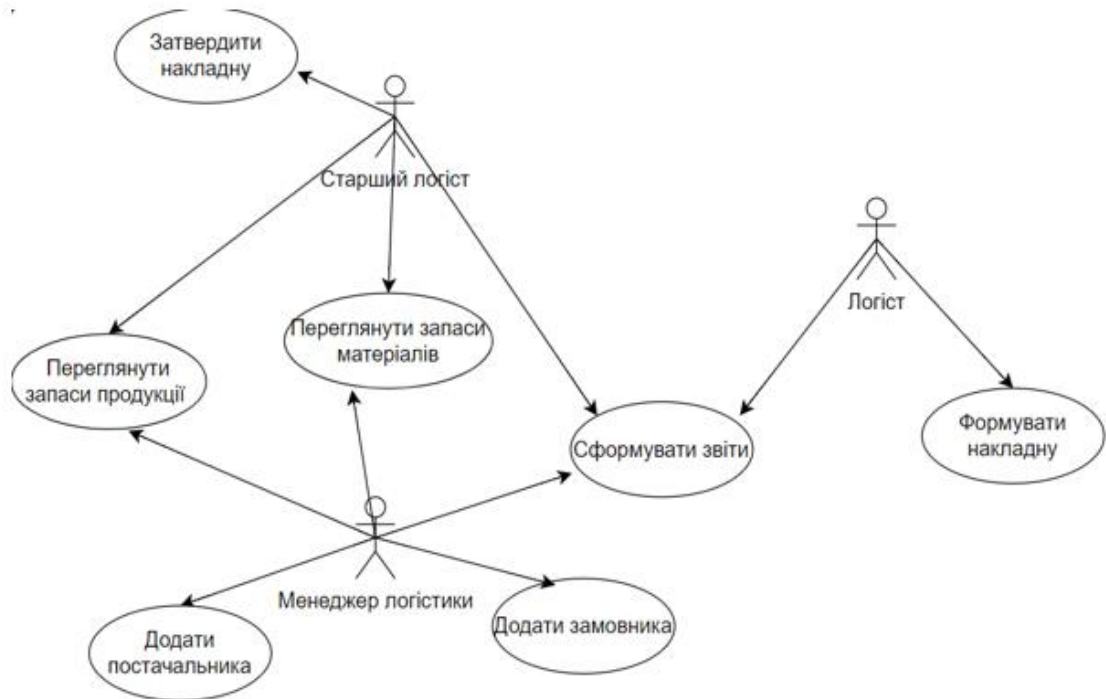


Найкоротший шлях комівояжера
через 15 міст Німеччини

В основу розробки інформаційної системи складського господарства пивзаводу покладено задачу комівояжера. Задача комівояжера (TSP, Travelling Salesman Problem) - це класична задача оптимізації, яка полягає в знаходженні найкоротшого маршруту, що проходить через кожну точку заданої множини точок лише один раз.

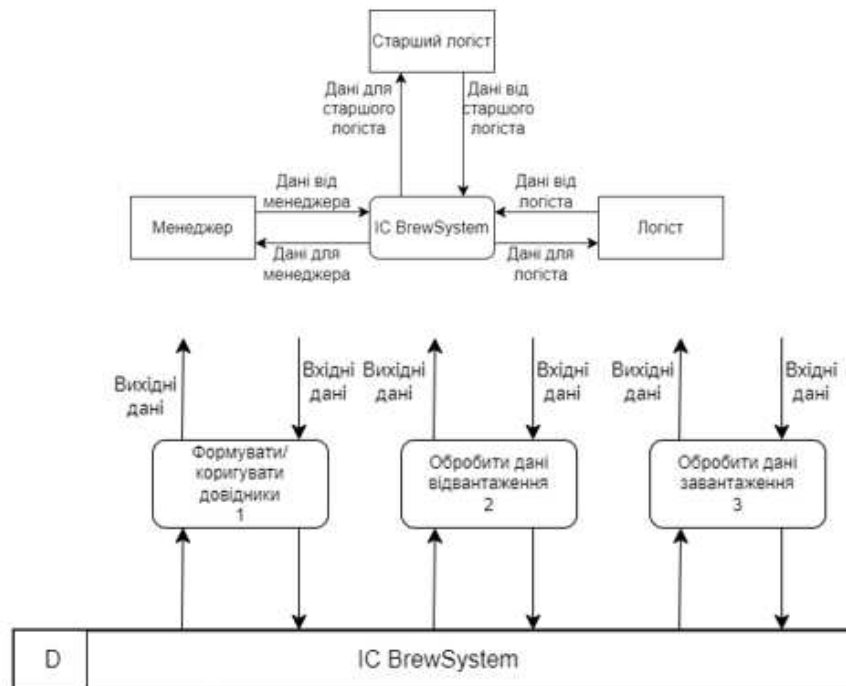
У контексті складського господарства пивзаводу, задача комівояжера може бути використана для розробки маршрутів для доставки сировини від постачальника до складу, а також для розробки маршрутів для доставки готової продукції зі складу до клієнтів.

Use-case діаграма



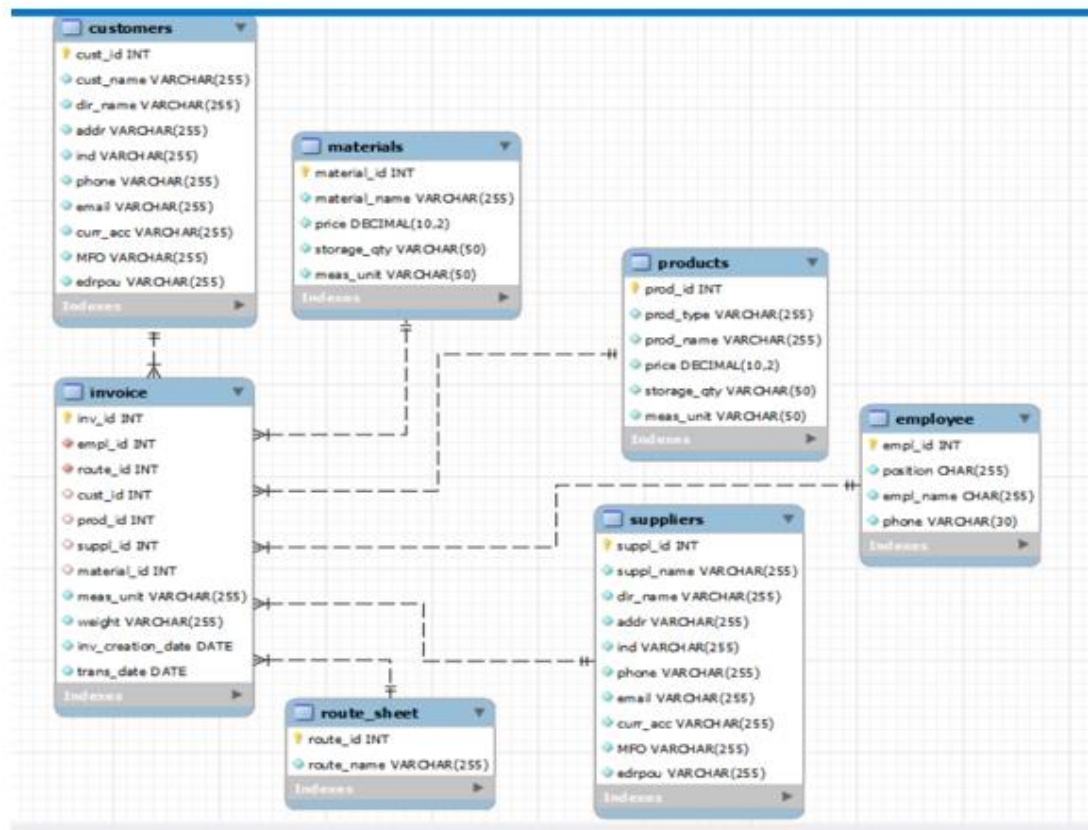
Слайд 6

Потоки даних (0-го та 1-го рівнів)

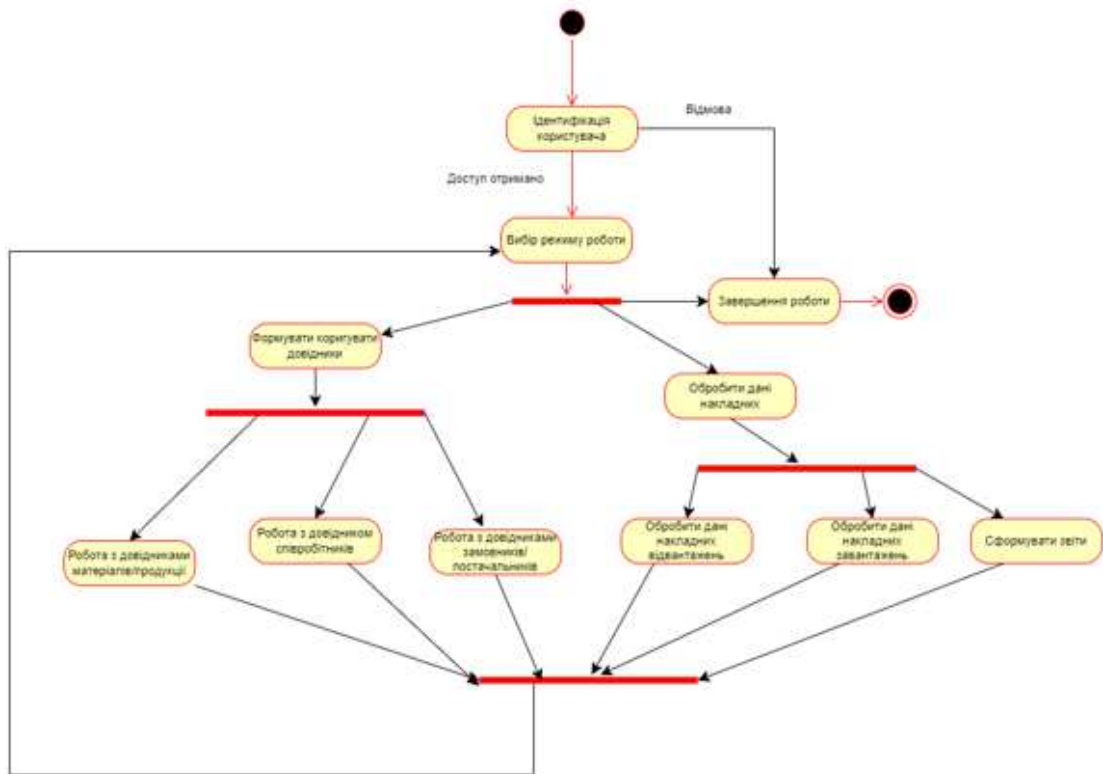


Слайд 7

Фізична модель БД



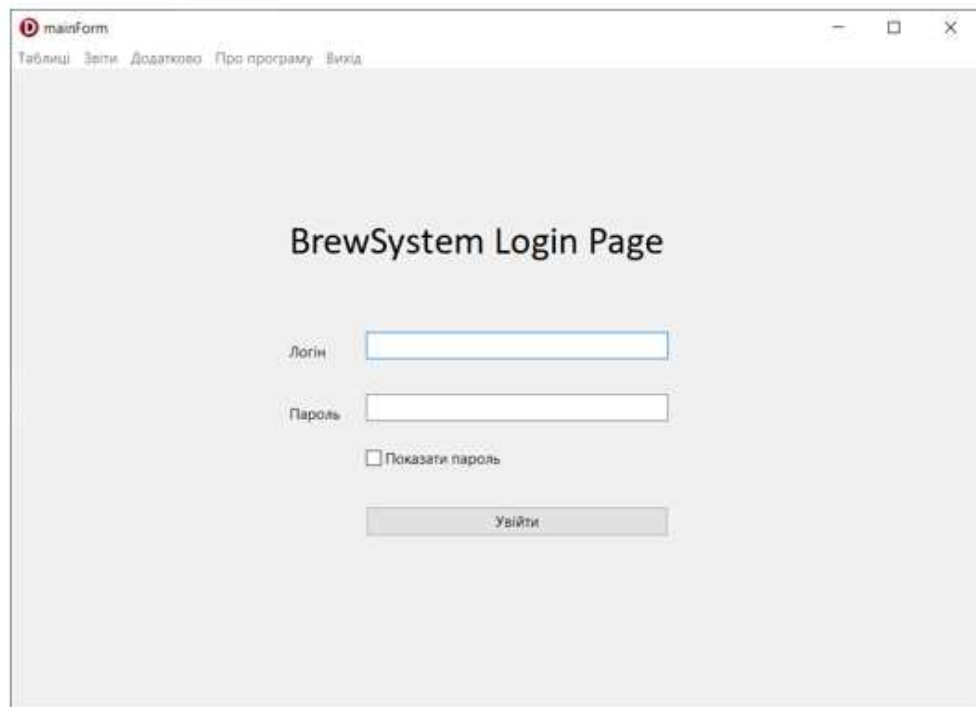
Діаграма станів



Слайд 9

Опис створеного програмного забезпечення

Форма входу в програму



The image shows a screenshot of a web application window titled "mainform". The window has a menu bar with the following items: "Таблиці", "Звіт", "Додатково", "Про програму", and "Вихід". The main content area is titled "BrewSystem Login Page". It contains a login form with the following elements:

- A label "Логін" (Login) next to a text input field.
- A label "Пароль" (Password) next to a text input field.
- A checkbox labeled "Показати пароль" (Show password).
- A button labeled "Увійти" (Login).

Слайд 10

Економічне обґрунтування

В основу техніко-економічного обґрунтування щодо необхідності впровадження нової інформаційної системи для складського господарства полягає в розробці заходів з інформаційної підтримки підприємства харчової промисловості в умовах наслідків часткової руйнації та дефіциту прісної води у місті Миколаєві. Що в свою чергу дозволить пришвидшити запуск виробництва та ефективне управління складським господарством

Висновок

- Зроблено наступні висновки
- 1) Проаналізовано особливості процесів з організації управління складським господарством пивзаводу та сучасний стан інформаційного забезпечення; визначено потенційні системи, які можуть бути введені та використовуватися для управління складом.
- 2) Запропоновано основні принципи розробки інформаційної системи складського господарства пивзаводу, в основу проектування якої покладено задачу комівояжера з складанням технічного та робочого проектів.
- 3) На мові програмування Delphi розроблено основні проектні рішення щодо програмного забезпечення, створено фізичну модель бази даних.
- 4) Розглянуто заходи щодо підвищення економічних показників, розраховано та визначено строк окупності та кошторис на виробництво і впровадження.
- 5) Проаналізовано фактори ризику, які можуть виникати під час роботи за персональним комп'ютером та розглянуто заходи щодо охорони навколишнього середовища при утилізації комп'ютерного обладнання.

Наукова новизна

- **Наукова новизна** роботи полягає в наступному:
- вдосконалено підхід до управління процесами обробки і систематизації даних про продукцію, сировину, споживачів, документацію, що на відміну від існуючих рішень, дозволяє підвищити ефективність виконання розв'язання логістичних задач.
- вперше для складського господарства пивзаводу запропоновано теоретично-практичне підґрунтя кординації цілей з розширенням можливостей та виходом на міжнародний ринок в умовах воєнного стану.

Апробація результатів досліджень



Основні положення і результати роботи доповідалися на XXXIX-й Міжнародній науково-практичній конференції «Сучасні аспекти модернізації науки: стан, проблеми, тенденції розвитку» (07 грудня 2023 року, м. Гетеборг, Швеція, дистанційно)