



## НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних системКафедра Комп'ютерні наукиОсвітній ступінь бакалаврСпеціальність 122 «Комп'ютерні науки»

(код і назва)

Освітньо-професійна програма Комп'ютерні науки

(назва)

## ЗАТВЕРДЖУЮ

Завідувач

кафедри Інформаційних технологій,  
штучного інтелекту і кібербезпекиГрибков Сергій Віталійович« 15 » квітня 2024 року

## ЗАВДАННЯ

Булія Даніла Юрійовича

(прізвище, ім'я, по батькові)

1. Тема роботи: Розроблення інформаційної системи обліку готової продукції на складі ПрАТ «Оболонь»Керівник роботи Харкянен Олена Валеріївна, к.т.н., доцент

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

затверджені наказом закладу вищої освіти від « 15 » квітня 2024 р. № 2792. Строк подання здобувачем роботи: 29.05.20243. Вихідні дані до роботи: Номенклатура продукції, список замовників, перелік складів, списки водіїв, перелік автотранспорту, шаблони замовлень з оформленими pdf-файлами4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):  
системний аналіз діяльності підприємства; функціональна модель діяльності складу підприємства: статус AS-IS та TO-BE; розроблення інтерфейсу користувача; налаштування функціоналу системи; написання інструкції користувача, розроблення заходів з охорони праці та навколишнього середовища.

5. Перелік графічного матеріалу:

Організаційна модель підприємства, функціональна модель діяльності складу готової продукції підприємства: статус AS-IS та TO-BE, логічна модель БД, фізична модель БД, схема бази даних в MS SQL Server, скріншоти інтерфейсу інформаційної системи



## АНОТАЦІЯ

Кваліфікаційна робота містить 89 сторінок, 64 рисунків, 10 таблиць, 6 додатків та 10 джерел.

Перший розділ містить загальну характеристику підприємства, на основі якого розроблялась система. У ньому досліджено організаційну структуру, проведено аналіз поточного стану автоматизації, розроблено список необхідних функцій і визначено задачу на автоматизацію. Також проведено аналіз програм-аналогів для виявлення їх недоліків та обґрунтування необхідності розробки нової системи.

Другий розділ присвячений технічному завданню на проектування інформаційної системи. У ньому визначено вимоги, специфікації та очікування від майбутньої системи.

Третій розділ описує процес створення системи. Тут складено інструкцію користувача, наведено перелік використаних технологій та детально описано процес розробки функцій системи. Крім того, проведено техніко-економічне обґрунтування та розраховано термін окупності системи.

Четвертий розділ розглядає питання охорони праці та захисту навколишнього середовища.

**КЛЮЧОВІ СЛОВА:** СКЛАД, ERWIN, PostgreSQL, TKINTER, PYTHON, ІНФОРМАЦІЙНА СИСТЕМА

## SUMMARY

The qualification work contains 89 pages, 64 figures, 10 tables, 6 appendices and 10 sources.

The first section contains a general description of the enterprise on the basis of which the system was developed. It examines the organizational structure, analyzes the current state of automation, develops a list of necessary functions, and defines the task of automation. Analogous programs were also analyzed to identify their shortcomings and justify the need to develop a new system.

The second section is devoted to the technical task of designing an information system. It defines the requirements, specifications and expectations for the future system.

The third section describes the system creation process. The user manual is compiled here, the list of technologies used is given, and the process of developing the system functions is described in detail. In addition, a technical and economic feasibility study was conducted and the payback period of the system was calculated.

The fourth chapter deals with issues of labor protection and environmental protection.

**KEYWORDS:** WAREHOUSE, ERWIN, PostgreSQL, TKINTER, PYTHON, INFORMATION SYSTEM

## ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ОБ'ЄКТУ АВТОМАТИЗАЦІЇ ТА ПОСТАНОВКА ЗАДАЧІ НА ПРОЕКТУВАННЯ.....	9
1.1 Загальна характеристика ПрАТ «Оболонь».....	9
1.2 Організаційна структура ПрАТ «Оболонь», роль і взаємодія підрозділів.....	11
1.3 Аналіз нинішнього стану комп'ютеризації ПрАТ «Оболонь» .....	17
1.4 Розроблення функціональної моделі та аналіз існуючих бізнес-процесів.....	19
1.5 Огляд існуючих рішень для розв'язання виявлених проблем .....	20
1.6 Обґрунтування доцільності проектування й розроблення інформаційної системи для обліку продукції на складі ПрАТ «Оболонь».....	26
1.7 Концептуальна модель системи .....	27
1.8 Розрахунок економічного ефекту від впровадження системи .....	27
РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЕКТУВАННЯ .....	36
2.1. Загальні положення.....	36
2.2. Призначення та цілі створення системи .....	36
2.3. Характеристика об'єкта автоматизації.....	36
2.4. Вимоги до системи.....	37
2.5. Склад і зміст робіт по створенню системи .....	50
2.6. Порядок контролю і приймання системи для інформаційної системи обліку готової продукції на складі ПрАТ «Оболонь».....	50
2.7. Вимоги до складу і змісту робіт із підготовки до введення системи в дію для інформаційної системи обліку готової продукції на складі ПрАТ «Оболонь».....	51
2.8. Вимоги до документації для інформаційної системи обліку готової продукції на складі ПрАТ «Оболонь».....	51
2.9. Джерела розробки для інформаційної системи обліку готової продукції на складі ПрАТ «Оболонь» .....	51
РОЗДІЛ 3. ОПИС КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ .....	53
3.1. Інформаційне забезпечення системи.....	53
3.2. Алгоритмізація та реалізація комплексу задач автоматизації .....	57
3.3. Інструкція користувача .....	71
РОЗДІЛ 4. ОХОРОНА ПРАЦІ.....	84
4.1. Охорона праці на підприємстві .....	84

ВИСНОВОК .....	86
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	87
ДОДАТКИ .....	90

## ВСТУП

В сучасному світі, де технологічний прогрес визначає конкурентоспроможність підприємств, впровадження інформаційних технологій стає необхідністю для ефективного управління та оптимізації бізнес-процесів.

Автоматизована інформаційна система (АІС) - це комплекс програмних і апаратних засобів, який призначений для збору, зберігання, обробки, передачі та використання інформації з метою автоматизації операцій та підтримки прийняття рішень у різних сферах діяльності. Ця система може включати в себе різноманітні компоненти, такі як бази даних, програмне забезпечення для обробки даних, сенсори для збору інформації, мережеве обладнання для передачі даних та інше.

Головна перевага автоматизованих інформаційних систем полягає в тому, що вони дозволяють автоматизувати рутинні операції, зменшуючи людський вплив на процеси та знижуючи ризик помилок.

Автоматизація складської діяльності за допомогою інформаційних систем має стратегічне значення для підприємства з огляду на кілька ключових аспектів. По-перше, це забезпечує ефективне управління запасами, дозволяючи точно відстежувати наявність товарів на складі та оптимізувати їх розподіл для максимізації ефективності. По-друге, автоматизація сприяє оптимізації внутрішніх процесів, зменшуючи час на виконання операцій та підвищуючи швидкість обробки замовлень. Третій аспект полягає в підвищенні точності та надійності обліку, що дозволяє уникнути помилок та мінімізувати ризик втрати або пошкодження товарів.

У кваліфікаційній роботі розроблялась інформаційна система для ПрАТ «Оболонь» з метою оптимізації та автоматизації роботи на складі. Головною метою її впровадження є забезпечення ефективного управління запасами, мінімізації помилок та часу виконання замовлень.

# РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ОБ'ЄКТУ АВТОМАТИЗАЦІЇ ТА ПОСТАНОВКА ЗАДАЧІ НА ПРОЕКТУВАННЯ

## 1.1. Загальна характеристика ПрАТ «Оболонь»

«Оболонь» є найбільшим виробником напоїв в Україні з 100% українським капіталом. Компанія спеціалізується на виробництві пива, безалкогольних та слабоалкогольних напоїв, артезіанських та питних вод. Вона також є лідером з експорту пива, що становить 62% всього експорту країни за 2021 рік. Завод "Оболонь" є найпотужнішим пивоварним підприємством в Європі. На даний момент "Оболонь" налічує головний завод у Києві та вісім підприємств у різних регіонах України.

У 1974 р. у новому районі Києва Оболоні на березі річки Почайни, що є притокою Дніпра, розпочалося будівництво майбутнього Київського пивзаводу № 3. Місце закладання підприємства було обрано не випадково. Проектування заводу здійснювали спеціалісти з Чехії. Вони ж і визначили, що м'яка та кришталево чиста вода юрського горизонту, величезні запаси якої було знайдено в цій місцевості, ідеально відповідало всім вимогливим критеріям пивоваріння. Цей фактор став вирішальним, і з того часу провідне українське харчове підприємство для виробництва своєї продукції використовує воду з власних артезіанських свердловин, що розташовані на території заводу, по теперішній час [1].

Вже у 1980 р. відбулося офіційне відкриття Київського пивзаводу № 3, що було приурочено до 22-х Олімпійських ігор, та випуск першої партії пива. На підприємстві був введений у дію цех розливу безалкогольної продукції, а через два роки пивзавод випустив свою першу продукцію — мінеральну воду [1].

У 1986 р. на базі Київського пивзаводу № 3 було створене пиво-безалкогольне об'єднання «Оболонь», до якого ввійшли також Київський пивзавод № 1, Київський пивзавод № 2 («Пивзавод на Подолі») та Фастівський пивзавод [1].

У 1989 р. на посаду Генеральним директором пиво-безалкогольного об'єднання «Оболонь» колектив обрав випускника НУХТ Олександра Слободяна. Підприємство першим у колишньому СРСР почало експортувати пиво за кордон.

Розпочалась експортна діяльність підприємства, і вперше «Жигулівське пиво» («Zhiguli Beer») виробництва «Оболоні» почало надходити до споживачів Великобританії, де отримало високу оцінку [1].

На початку 90-х років відбулася приватизація заводу. У 1990 р. колектив компанії підписав договір оренди з правом викупу. Цей період відзначився зростанням та розвитком брендів компанії, включаючи випуск першого українського бренду пива — ТМ "ОБОЛОНЬ".

Приватизоване підприємство отримало юридичний статус Закритого акціонерного товариства. Була введена в дію лінія розливу пива в пляшки ємністю 0,33 літра [1].

У 1994 р. ЗАТ "Оболонь" вперше розпочало виробництво слабоалкогольного напою "Джин Тонік", встановивши тим самим новий ринок для слабоалкогольних напоїв в Україні та Східній Європі. Компанія стала власником недобудованого об'єкта в м. Красилів Хмельницької області, який зараз функціонує як дочірнє підприємство, випускаючи під брендом "Оболонь" мінеральну воду, безалкогольні та слабоалкогольні напої. Перша оптово-торгівельна база ЗАТ "Оболонь" була створена в м. Красилів. Наразі компанія має дев'ятнадцять баз у різних регіонах країни, які реалізують продукцію компанії [1].

У 2000-х роках компанія продовжувала розширювати свій асортимент продукції та збільшувати свою присутність на ринку. Було випущено нові види пива та безалкогольних напоїв, а також розпочато експортування своєї продукції за кордон [1].

У 2010 р. корпорація "Оболонь" визначилася зі своєю інноваційною діяльністю, випускаючи нові продукти, проводячи редизайн брендів і розробляючи нові напої. Також компанія активно впроваджувала екологічні проекти, спрямовані на енергозбереження, безвідходні технології та екологічну безпеку [1].

У 2020 р. ПрАТ "ОБОЛОНЬ" відсвяткувала свій 40-річний ювілей, під час якого компанія відзначилася масштабним оновленням та розширенням свого асортименту продукції, випуском нових сортів пива, сидру, безалкогольних та інших напоїв [1].

З початком війни підприємству довелося виживати в умовах закриття роздрібних магазинів, заборони на торгівлю алкоголем та дефіцитом палива. Після введення воєнного стану, запровадженого наприкінці лютого 2022 р., було заборонено продаж алкоголю. У березні ПрАТ «Оболонь» випустила лише 6,5 % торішнього обсягу пива за цей же місяць.

Деякі виробничі майданчики, наприклад, Охтирський пивоварний завод, перейшли на виробництво питної води для місцевих лікарень та населення.

У такий непростий час найбільший виробник напоїв України продовжив розвиватися і, навіть, випустив новинку — енергетичний напій «ДЖЕТТ». Як зазначають у компанії JETT (ДЖЕТТ) — це високо потужний та функціональний безалкогольний енергетичний напій.

## **1.2. Організаційна структура ПрАТ «Оболонь», роль і взаємодія підрозділів**

Організаційна структура підприємства – це система взаємовідносин між різними підрозділами та рівнями управління, яка визначає порядок, ієрархію та розподіл обов'язків, повноважень і відповідальності між ними. Вона встановлює зв'язки та взаємодію між працівниками, сприяючи ефективному досягненню цілей і завдань підприємства.

### **1.2.1. Загальна схема організації структури**

Організаційну структуру ПрАТ «Оболонь» ілюстровано на рисунку А.1 додатку А.

Організаційна структура підприємства має вертикальну структуру управління, в якій чітко визначена ієрархія підпорядкування та відповідальності, що забезпечує централізоване прийняття рішень.

Структурні компоненти ПрАТ "Оболонь" включають такі елементи:

- президент корпорації "Оболонь" відповідає за стратегічне та оперативне планування;
- генеральний директор забезпечує стратегічне планування, оперативне управління та безпеку праці;
- директор з управління персоналом відповідає за організацію роботи з

персоналом, регулювання трудових відносин та розвиток персоналу, керує відділом кадрів;

- директор з продажу організовує роботу збуту готової продукції і керує відділом збуту;

- директор з маркетингу спеціалізується на стратегічному маркетинговому плануванні, розробці маркетингових кампаній та взаємодії з медіа-ресурсами, а також рекламі та зв'язкам з громадськістю, керує маркетинговим відділом;

- фінансовий директор відповідає за стратегічний аналіз, фінансове планування та управління корпоративною фінансовою структурою, керує фінансовим відділом;

- директор з корпоративних питань та розвитку займається розвитком підприємств з корпоративними правами, диверсифікацією бізнесу, розвитком фірмової торгівлі та мережі фірмових магазинів "О-маркет", керує відділом розвитку;

- директор з зовнішньо-економічних питань відповідає за зовнішньо-економічну діяльність, розвиток експортних ринків та операції з імпорту, керує відділом зовнішньо-економічних питань;

- директор з операційної діяльності здійснює технічний контроль якості продукції на всіх етапах виробництва, контролює функціонування систем управління ISO, керує відділом якості продукції;

- директор з матеріально-технічного постачання відповідає за управління та координацію централізованого постачання сировини та допоміжних матеріалів, а також реалізацію сировини власного виробництва, керує відділом логістики;

- директор з інформаційних технологій відповідає за впровадження інноваційних технологій, управління інформаційними ресурсами, безпеку та розвиток усіх інформатизованих процесів у компанії, а також управління інвестиціями в ІТ-сфері для підвищення продуктивності та впровадження інновацій, керує ІТ-відділом.

## 1.2.2. Структура ІТ-відділу ПрАТ «Оболонь»



Рисунок 1.2 - Організаційна структура ІТ- відділу ПрАТ «Оболонь»

ІТ-відділ підприємства відповідає за розробку, впровадження та підтримку інформаційних систем і технологій, забезпечує безпеку даних, технічну підтримку співробітників, бере участь у стратегічному плануванні розвитку компанії та впровадженні інновацій для підвищення ефективності роботи.

Розробимо функціональну модель бізнес-процесу діяльності відділу ІТ-відділ продемонструємо, що зображена у додатку Б (Рисунок Б.1 -)

### 1.2.2.1. Посадові обов'язки працівників

Спеціалісти з кібербезпеки відіграють ключову роль у захисті компанії від сучасних кіберзагроз, забезпечуючи безпеку та надійність інформаційних систем і даних.

Основні обов'язки спеціалістів з кібербезпеки:

1. Захист від кіберзагроз:
  - розроблення та впровадження заходів для запобігання кібератак, виявлення та виправлення інцидентів.
2. Аудит безпеки:
  - проведення регулярних аудитів та оцінок безпеки систем, мереж та додатків для виявлення потенційних вразливостей.
3. Впровадження заходів безпеки:
  - розроблення та впровадження політик безпеки, шифрування даних та інших технічних засобів захисту.

4. Моніторинг та реагування:
  - постійний моніторинг інформаційних систем для виявлення аномальних активностей та негайна реакція на потенційні загрози.
5. Навчання персоналу:
  - проведення навчань та інформаційних кампаній для персоналу щодо правил кібербезпеки та виявлення соціально-інженерних атак.
6. Взаємодія з іншими відділами:
  - співпраця з іншими відділами, зокрема з IT-відділом, для забезпечення високого рівня захисту всіх інформаційних ресурсів.
7. обговорення стратегій безпеки:
  - участь у розробці та обговоренні стратегій загальної безпеки компанії та планів реагування на кризові ситуації.

Програмні розробники в компанії "Оболонь" є фахівцями, відповідальними за розробку та підтримку програмного забезпечення. Основні завдання та обов'язки цих спеціалістів включають:

1. Розробку програмного забезпечення:
  - створення програм та додатків відповідно до вимог інших відділів та бізнес-потреб компанії.
2. Тестування та відлагодження:
  - проведення тестів для перевірки функціональності та виявлення можливих помилок, а також виправлення виявлених недоліків.
3. Оптимізацію та покращення:
  - вдосконалення та оптимізація існуючого програмного забезпечення для забезпечення оптимальної продуктивності та ефективності.
4. Впровадження нових функцій:
  - додавання нових функцій та можливостей до програм для відповіді на зміни в бізнес-вимогах.
5. Співпрацю з іншими відділами:
  - взаємодія з іншими відділами, зокрема з маркетингом, продажами для забезпечення відповідності програмного забезпечення бізнес-потребам.

6. Документація коду:

- створення та підтримка документації коду для полегшення розуміння та співпраці з іншими розробниками.

7. Підтримка та вдосконалення:

- надання технічної підтримки та вдосконалення програм компанії

Системні адміністратори в компанії "Оболонь" є фахівцями, відповідальними за ефективне функціонування та управління інформаційними системами та мережами компанії. Основні завдання та обов'язки цих спеціалістів включають:

1. Управління серверами:

- налаштування, моніторинг та підтримка серверів для забезпечення їхньої надійності та продуктивності.

2. Мережеве адміністрування:

- конфігурація та управління мережевою інфраструктурою для забезпечення стабільної роботи мережі.

3. Безпека інформаційних систем:

- впровадження заходів для захисту від несанкціонованого доступу, вірусів та інших кіберзагроз.

4. Резервне копіювання та відновлення:

- забезпечення регулярного резервного копіювання даних та розроблення планів відновлення в разі аварій.

5. Підтримка електронної пошти та комунікацій:

- конфігурація та підтримка електронної пошти, відеоконференцій та інших комунікаційних сервісів.

6. Відслідковування пристроїв та обладнання:

- моніторинг та керування обладнанням, відстеження його стану та регулярна підтримка.

7. Співпраця з іншими відділами:

- взаємодія з різними відділами, зокрема з розробкою програмного забезпечення та кібербезпекою, для забезпечення інтеграції та взаємодії інформаційних систем.

#### 8. Технічна підтримка:

- надання технічної підтримки користувачам та вирішення технічних проблем.

#### 1.2.2.2. Взаємодія ІТ-відділу з іншими структурними підрозділами підприємства

Оскільки в процесі діяльності, працівники відділів зобов'язані обмінюватись даними з іншими підрозділами нижче представлено таблицю взаємодії підрозділів, які будуть користувачами інформаційної системи. В таблиці наведено інформаційні та матеріальні потоки, що надходять в досліджуваній відділ.

Таблиця 1.1. Вимоги до технічного забезпечення відділу

Назва підрозділу	Отримує інформацію від ІТ-відділу	Передає інформацію до ІТ-відділу
HR-відділ	Системи управління персоналом, HR-аналітика, технічна підтримка	Інформація про співробітників, рекрутинг, вимоги до систем управління персоналом
Фінансовий відділ	Програмне забезпечення для фінансового обліку, аналізу даних, стратегічного планування	Фінансові звіти, бюджети, запити на впровадження фінансових програм
Відділ маркетингу	Дані про користувачів, аналітика веб-сайту, інструменти аналізу даних	Запити на створення веб-сайту, CRM системи, автоматизовані маркетингові кампанії

*Продовження таблиці 1.1. Вимоги до технічного забезпечення відділу*

Назва підрозділу	Отримує інформацію від ІТ-відділу	Передає інформацію до ІТ-відділу
Відділ збуту і постачання	Інформація про замовлення, запаси, системи управління запасами, автоматизація замовлень	Запити на розробку систем управління запасами, оптимізацію збутових операцій
Відділ розвитку	Технологічні інновації, інформація про можливості та обмеження технічних рішень	Вимоги до розробки нових продуктів, технологій
Відділ виробництва продукції	Програмні рішення для автоматизації контролю якості, виробничих операцій, моніторингу обладнання	Вимоги до нових програм, оновлення програмного забезпечення

### **1.3. Аналіз нинішнього стану комп'ютеризації ПрАТ «Оболонь»**

На ПрАТ "Оболонь" широко використовуються автоматизовані програми для оптимізації та підвищення продуктивності бізнес-процесів.

У сфері взаємодії з клієнтами впроваджена CRM-система, що дозволяє ефективно відстежувати інформацію про клієнтів, їхні замовлення та взаємодію з представниками компанії.

Оптимізація управління запасами та складськими операціями відбувається за допомогою спеціалізованої складської програми, яка спрощує відстеження руху товарів, управління запасами та автоматизацію процесів замовлення і постачання.

Для фінансового управління застосовуються спеціалізовані програми, що об'єднують бюджетування, фінансовий аналіз і стратегічне планування.

У сфері управління персоналом використовується HR-система, яка автоматизує електронний облік працівників, планування навчань, відстеження розвитку кадрів та інші функції.

Крім того, автоматизація охоплює й виробничі процеси, такі як контроль якості, системи моніторингу виробництва та інші технології, спрямовані на підвищення ефективності виробництва напоїв.

Також впроваджено власноруч розроблену ERP-систему, яка використовується як веб-сервіс на основі Javascript та фреймворків цієї мови. Основа системи — це надійна база даних Oracle, що забезпечує зберігання та обробку великого обсягу інформації. Створена ERP система охоплює різні аспекти діяльності компанії, включаючи фінанси та виробництво.

Крім того підприємство інтегрується з бензоколонками великих компаній за допомогою API, що дозволяє відстежувати витрати на паливо і взаємодіяти з активностями водіїв доставки та логістики.

Для організації корпоративної комунікації та обміну файлами між співробітниками використовується система Microsoft Exchange. Ця система також містить інструменти для обробки інформації та документів.

Додаткові системи, такі як HDS (E-travel and Expense Notes) та HDS (PoA), відповідають за обробку електронних записів для співробітників та підтвердження довіреностей відповідно.

Щодо автоматизації виробничих процесів, на підприємстві використовується система штрих-кодів для управління запасами, логістики та обміну документами з системою SAP.

Аналіз поточного стану автоматизації на підприємстві свідчить про наявність автоматизованих систем, проте деякі з них вже застарілі та потребують модернізації для забезпечення нового функціоналу для ефективної роботи та відповідності сучасним вимогам. Також оновлення приведе до покращення не тільки продуктивності, а й до покращення безпеки даних підприємства.

## **1.4. Розроблення функціональної моделі та аналіз існуючих бізнес-процесів**

Функціональна модель інформаційної системи (ІС) - це уявлення про роботу та взаємодію складових системи на рівні функцій та процесів, що вона виконує. Основна мета такої моделі - зрозуміти, як саме система функціонує, які операції вона виконує, та як вона взаємодіє з іншими системами чи користувачами.

### **1.4.1. Функціональна модель діяльності складу готової продукції**

Функціональна модель діяльності складу готової продукції побудована в нотації BPMN програмного засобу Bizagi і має два рівні декомпозиції. В додатку Б (0) продемонстровано загальний бізнес-процес даної ІС.

Алгоритм роботи ІС складається з таких блоків:

1. Прийом продукції з документами.
2. Розвантаження товару вантажниками.
3. Перевірка кількості і цілісності продукції.
4. Визначення зон і умов зберігання продукції.
5. Переміщення товару до встановлених зон зберігання.
6. Процес руху готової продукції на складі (управління продукцією).  
Декомпозиція процесу наведена в додатку Б (Рисунок Б.3 -).
7. Підготовка продукції до відпуску, до її відправлення.
8. Перевірка наявності і цілісності товару.
9. Комплектація товару.
10. Оформлення товаро-транспортної накладної.

Блок руху готової продукції на складі також має свій алгоритм підпроцесу:

- 1) Функція списання продукції або занесення даних щодо прийнятої продукції або інвентаризація.
- 2) Обрахунок залишків.

### **1.4.2. Виявлені проблеми**

У ході аналізу автоматизованих програм на підприємстві, було визначено ряд проблем, які потребують вирішення у ході створення інформаційної системи:

- обмежена функціональність: застарілі системи обмежені у функціональності, що може ускладнювати виконання сучасних бізнес-процесів та вимог клієнтів;
- низька продуктивність: застарілі системи можуть працювати повільно та неефективно, що призводить до затримок у виконанні завдань та втрати часу працівників;
- несумісність з сучасними технологіями: застарілі системи можуть бути несумісними з сучасними програмами та технологіями, що ускладнює їх інтеграцію з іншими системами та використання новітніх інструментів;

- низька безпека: не впроваджені системи входу до інформаційних систем.

### **1.4.3. Задачі автоматизації**

Шляхом виявлення функцій підприємства, які потребують автоматизації, було виявлено ряд задач для подальшої розробки та автоматизації:

- оптимізація бізнес-процесів: впровадження або оновлення систем управління бізнес-процесами, які дозволяють автоматизувати та оптимізувати послідовність операцій;
- додавання системи входу до інформаційної системи для забезпечення безпеки інформації на підприємстві;
- автоматизація рутинних завдань: розроблення програмного забезпечення для автоматизації рутинних операцій та процесів, що звільняє час працівників для виконання більш важливих завдань;
- оновлення інформаційних систем: впровадження нових інформаційних систем на популярній мові програмування з новим інтерфейсом і оптимізованими функціями для підвищення ефективності та конкурентоспроможності підприємства.

### **1.5. Огляд існуючих рішень для розв'язання виявлених проблем**

В сучасному світі, де швидкість розвитку технологій постійно зростає, розроблення програмного забезпечення стає все більш важливою для багатьох сфер життя, включаючи бізнес, освіту, медицину та інші. З цим пов'язане підвищення попиту на ефективні та продуктивні методи розробки програмного забезпечення.

Один з важливих етапів розробки програмного забезпечення - це аналіз існуючих аналогів інформаційної системи (ІС), що дозволяє розробникам отримати інсайди щодо найкращих практик, технік та інструментів, що використовуються у галузі. Аналіз існуючих аналогів ІС допомагає виявити переваги та недоліки різних підходів до розробки програмного забезпечення, а також визначити можливі напрямки для вдосконалення власного процесу розробки

### **1.5.1. ІС: Торгівля і склад**

ІС: Підприємство — це програма, завданням якої є автоматизація процесів бухгалтерського, управлінського, податкового, кадрового обліку та, як наслідок, оптимізація використання всіх ресурсів підприємства. Це не новий діджитал-винахід: адже цьому ПЗ більше ніж двадцять років, і за цей час воно стало одним з головних інструментів малого, середнього бізнесу, бюджетних, державних структур, а також великих холдингів і підприємств.

Функції:

- створення та керування товарами;
- управління запасами;
- облік поставчань та замовлень;
- управління ціноутворенням;
- контроль термінів придатності;
- аналітика та звітність;
- керування персоналом;
- керування клієнтами;
- фінансовий облік;
- управління рахунками і платежами;
- управління податками;
- робота зі штрих-кодом.

### **1.5.2. Oracle Warehouse Management**

Oracle Warehouse Management (OWM) — це програмне забезпечення, розроблене компанією Oracle, яке призначене для управління складськими

операціями, такими як отримання, зберігання і відвантаження товарів. Ця система допомагає підприємствам ефективно виконувати складські процеси, автоматизувати операції та оптимізувати використання складського простору/

Функції:

- створення та керування товарами;
- управління запасами;
- облік поставок та замовлень;
- управління ціноутворення;
- аналітика та звітність;
- управління рахунками і платежами;
- система безпеки;
- інтеграція з іншими системами;
- робота зі штрих-кодом.

### **1.5.3. Microsoft Dynamics 365**

Microsoft Dynamics 365 — це сімейство інтегрованих хмарних застосунків для управління різними аспектами бізнесу, яке розробляється й утримується корпорацією Microsoft. Ця платформа надає комплексні рішення для автоматизації різних бізнес-процесів

Функції:

- створення та керування товарами;
- управління запасами;
- облік поставок та замовлень;
- управління ціноутворенням;
- аналітика та звітність;
- контроль термінів придатності;
- управління рахунками і платежами;
- управління контактами;
- управління зверненнями;
- система безпеки;

- робота зі штрих-кодом;
- інтеграція з іншими системами.

#### **1.5.4. Zoho Inventory**

Zoho Inventory — це програмне рішення для управління запасами та складським обліком, розроблене компанією Zoho Corporation. Ця платформа дозволяє бізнесам ефективно відслідковувати та керувати своїми запасами, проводити управління замовленнями та постачаннями, контролювати вартість товарів, а також аналізувати різноманітні дані про продажі та запаси для прийняття управлінських рішень.

Функції:

- створення та керування товарами;
- управління запасами;
- облік постачань та замовлень;
- управління ціноутворенням;
- контроль термінів придатності;
- система безпеки;
- інтеграція з іншими системами;
- робота зі штрих-кодом.

### 1.5.5. Порівняння систем-аналогів

Таблиця 1.2. Порівняння програм-аналогів складських ІС

Функція	Програми			
	1С: Торговля і склад	Oracle Warehouse Management	Microsoft Dynamics 365	Zoho Inventory
Створення та керування товарами	+	+	+	+
Управління запасами	+	+	+	+
Облік поставчань та замовлень	+	+	+	+
Управління ціноутворенням	+	+	+	+
Контроль термінів придатності	+	+	+	-
Аналітика та звітність	+	+	+	-
Керування персоналом	+	+	+	-
Керування клієнтами	+	+	+	-
Фінансовий облік	+	+	+	-
Управління рахунками і платежами	+	+	+	-

Продовження таблиці 1.2. Порівняння програм-аналогів складських ІС

Функція	Програми			
	1С: Торговля і склад	Oracle Warehouse Management	Microsoft Dynamics 365	Zoho Inventory
Аналітика та прогнозування	+	+	+	-
Фінансова звітність	+	+	+	-
Управління контактами	+	-	-	-
Управління зверненнями	+	-	-	-
Управління податками	+	-	-	+
Система безпеки	+	+	+	+
Інтеграція з іншими системами	+	+	+	+

На основі представлених в таблиці даних можна зробити висновок, що програма «1С: Торговля і склад» має найбільш розгалужений функціонал у порівнянні з іншими програмами. Програми «Oracle Warehouse Management» і «Microsoft Dynamics 365» відрізняються більше обмеженим функціоналом і не придатна для управління контактами, зверненнями і податками. Програма «Zoho Inventory» не підтримує функції щодо контролю термінів придатності, керування персоналом, клієнтами, фінансового обліку та фінансової звітності, управління рахунками, платежами, контактами, зверненнями, аналітики та прогнозування.

Вибір інформаційної системи для управління складом залежить від ряду інших факторів, включаючи бюджет компанії та конкретні потреби бізнесу. Кожна програма має свою вартість для впровадження і підтримки програми, тому важливо

обрати рішення, яке відповідає фінансовим можливостям компанії. Крім того, важливо враховувати специфічні потреби та вимоги бізнесу. Наприклад, якщо компанія потребує інтеграції з іншими системами або особливий функціонал, такий як аналітика або керування персоналом, ви можете обрати програму, яка найкраще відповідає цим вимогам.

### **1.6. Обґрунтування доцільності проектування й розроблення інформаційної системи для обліку продукції на складі ПрАТ «Оболонь»**

Кожне підприємство, враховуючи свої унікальні потреби та специфіку діяльності, має свої власні вимоги до програмного забезпечення. При встановленні програми для управління складом, також як і у будь-якій іншій області, підприємство розглядає різні аспекти, такі як функціональність, ефективність, вартість, можливості інтеграції, а також можливості розвитку та підтримки.

Впровадження нової інформаційної системи для обліку продукції на складі ПрАТ «Оболонь» є необхідним кроком для покращення ефективності операцій та підвищення точності обліку. Нова система дозволить оптимізувати процеси управління та контролю за продукцією, спростить роботу персоналу та зменшить ризик помилок. Вона також забезпечить надійне ведення обліку, що допоможе уникнути втрат та забезпечить точну інформацію для прийняття управлінських рішень. Враховуючи той факт, що програма розробляється на популярній мові програмування з легкою інтегрованістю з іншими системами та володіє розширеними графічними можливостями, впровадження нових функцій у систему документообігу буде проводитися ефективно та швидко. Це дозволить користувачам отримати доступ до потужних інструментів для оптимізації робочих процесів і значно покращить якість документообігу.

Після ретельного аналізу існуючих аналогів програмного забезпечення для управління складськими процесами ми прийшли до висновку, що жоден з розглянутих варіантів не відповідає повністю потребам та вимогам нашого підприємства. Значна частина аналогів програмного забезпечення, які були розглянуті, мають значну вартість впровадження та підтримки. деякі з цих аналогів мають обмежений функціонал, що не враховує всіх аспектів нашої діяльності.

Гарною інвестицією в майбутнє підприємства є створення власної інформаційної системи.

### **1.7. Концептуальна модель системи**

Виявлені в процесі моделювання діяльності складу недоліки буде виправлено при створенні моделі ТО-ВЕ (як буде) - моделі нової організації процесів.

Проаналізувавши роботу підприємства, було виявлено недоліки бізнес-процесу, а саме відсутність системи верифікації при вході до інформаційної системи (ІС). Для вирішення цієї проблеми було вирішено додати систему входу до ІС. Всі внесені змінні відображаються в додатку Б (Рисунок Б.4 -), а саме доданий блок «Реєстрація надходження товару» із системою входу до інформаційної системи.

Декомпозиція процесу реєстрації надходження товару наведена в додатку Б (Рисунок Б.5 -)

Додавання системи входу до інформаційної системи для зберігання даних в модель ТО-ВЕ є важливим кроком у забезпеченні безпеки та конфіденційності інформації на підприємстві. Ця система входу забезпечує контрольований доступ до інформації, дозволяючи лише авторизованим користувачам отримувати доступ до конфіденційних даних та функціональних можливостей системи. Такий підхід допомагає уникнути несанкціонованого доступу до даних та забезпечує відповідність вимогам щодо захисту інформації та регуляторних стандартів.

### **1.8. Розрахунок економічного ефекту від впровадження системи**

Розрахунок економічного ефекту від впровадження системи включає в себе оцінку фінансових вигід, які підприємство може отримати внаслідок удосконалення своїх бізнес-процесів за допомогою нової системи.

#### **1.8.1. Визначення розміру оплати праці**

Джерелами прибутку від впровадження системи обліку готової продукції можуть бути такі фактори:

- ефективний облік та контроль готової продукції;
- мінімізація часу інвентаризації;
- оптимізація управління зберіганням та переміщенням готової продукції;

- зниження витрат на облік та управління готовою продукцією;
- забезпечення надійності та точності даних обліку.

Визначаємо ознаку – тестування.

Ступінь новизни розроблюваних задач — "В" — використання типових проектних рішень за умови їх змін, розроблення проектів, що мають аналогічні рішення.

Група складності алгоритму — 3.

Узагальнені дані вхідної та вихідної інформації для системи обліку готової продукції на складі за видами вхідної та вихідної інформації таблиці 1.3

Таблиця 1.3. Узагальнені дані для вхідної та вихідної інформації системи  
«Підтримка роботи складу готової продукції»

Вид інформації	Позначення	К-сть наборів даних
Змінна інформація	ЗІ	m=5
Нормативно – довідкова інформація	НДІ	n=4
Банк(база) даних	БД	p=3
Обробка в режимі реального часу	РЧ	Так
Забезпечення телекомунікаційної обробки даних і управління віддаленими об'єктами	ТОУ	Ні

Таблиця 1.4. Визначення витрат часу для системи підтримки роботи складу готової продукції

Вид системи	Стадія розробки системи	
	Ескізний проект	Технічне завдання
	В	В
Облік готової продукції на складі. Управління та контроль за зберіганням, обліком та видачею продукції; оптимізація складських процесів.	$T_1=75$	$T_2=50$

Визначимо витрати часу на стадіях «технічний проект», «робочий проект» і «впровадження».

Вхідними даними для визначення є:

- кількість форм вхідної інформації 6;
- кількість форм вихідної інформації 4;
- базове значення витрат часу для стадії «Технічний проект» ТБ3=85;
- базове значення витрат часу для стадії «Робочий проект» ТБ4=100;
- базове значення витрат часу для стадії «Впровадження» ТБ5=50.

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

Розрахунок витрат часу для стадії «Технічний проект» системи підтримки обслуговування клієнтів ресторану.

Коефіцієнт трудомісткості робіт  $K_n$  визначається за формулою (1.1):

$$K_n = \frac{k_1 * m + k_2 * n + k_3 * p}{m + n + p} \quad (1.1)$$

$$k_n = \frac{(1 * 6 + 0.72 * 2 + 2.08 * 1)}{(6 + 4 + 1)} = 9.52/11 = 0.86$$

Таблиця 1.5. Коефіцієнти  $k_1$ ,  $k_2$ ,  $k_3$  для стадії «технічний проект»

Вид використаної інформації	Ступінь новизни
	В
$k_1$ (ЗІ)	1.0
$k_2$ (НДІ)	0,72
$k_3$ (БД)	2.08

Таблиця 1.6. Коефіцієнт ступеню новизни проекту,  $k_0$ 

Стадія розробки системи	Вид обробки	Ступінь новизни
		Г
Технічний проект	РЧ	1.26
Робочий проект	РЧ	1.32
Впровадження	РЧ	1.21

Витрати часу для стадії «Технічний проект» розраховуються за формулою (1.2):

$$T_3 = TЗ = ТБЗ * K_{\Pi} * K_0 = 85 * 0.86 * 1.26 = 92 \quad (1.2)$$

Таблиця 1.7. Коефіцієнти  $k_1$ ,  $k_2$ ,  $k_3$  для стадії «Робочий проект»

Вид використаної інформації	Ступінь новизни
	В
$k_1(ЗІ)$	1.0
$k_2(НДІ)$	0.50
$k_3(БД)$	0.35

$K_{\Pi}$  – коефіцієнт, що враховує вид інформації і визначається за формулою (1.3):

$$k_{\Pi} = \frac{k_1 * m + k_2 * n + k_3 * p}{m + n + p} \quad (1.3)$$

$$k_{\Pi} = \frac{(1.0 * 4 + 0.50 * 2 + 0.35 * 1)}{(6 + 4 + 1)} = 5.35 / 11 = 0.486$$

Витрати часу на стадії «робочий проект» ( $T_4$ ) розраховують за формулою (1.4):

$$T_4 = T_{B4} * k_{\Pi} * k_o * k_c \quad (1.4)$$

Для знаходження  $k_c$  для формули необхідно ідентифікувати складність контролю вхідної та вихідної інформації.

Тобто  $k_c = 1.3$

$$T_4 = 100 * 0.486 * 1.32 * 1.3 = 83$$

Визначення витрат часу на стадії «впровадження» ( $T_5$ ) вираховуємо за формулою (1.5):

$$T_5 = T_{B5} * k_{\Pi} * k_o * k_c \quad (1.5)$$

$$T_5 = 50 * 0.486 * 1.21 * 1.3 = 38$$

Отже, загальні витрати людської праці складають за формулою (1.6):

$$T_{\Sigma} = T_1 + T_2 + T_3 + T_4 + T_5 \quad (1.6)$$

$$T_{\Sigma} = 75 + 50 + 92 + 83 + 38 = 338$$

Визначимо чисельність виконавців  $\Psi$  за формулою (1.7).

$$\Psi = \frac{T_{\Sigma}}{\Phi} \quad (1.7)$$

Якщо для виконання курсової роботи припустимо кількість робочих годин складає 530 із 7-годинним робочим днем, тому на розробку проекту виділено  $\Phi$ , днів:

$$\Phi = 530/7 = 75 \text{ днів}$$

Для курсової роботи  $\Phi = 75$  днів. Тоді визначаймо кількість місяців із розрахунку 25 робочих днів.

Кількість місяців на розробку ( $M$ ) розраховуємо за формулою (1.8):

$$M = \Phi/25 \quad (1.8)$$

$$M = 75/25 = 3 \text{ місяці}$$

Отже, для виконання такого проекту потрібно така чисельність виконавців  $\Psi$ , яка обраховується за формулою:  $\Psi = 338/75 = 5$  виконавців

Приймемо розмір заробітної плати програміста - 25000 грн, тоді загальна сума заробітних плат програмістів складає:

$$V'_1 = Ч * М * ЗП = 5 * 3 * 25000 = 375000 \text{ грн} \quad (1.9)$$

### 1.8.2 Витрати, пов'язані з розробкою програми на ПК. Витрати на придбання і установку ПК

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

$$T_{\text{ПК}} = 2000 - (6 * 8 + 5 * 12) = 1892 \text{ год.} \quad (1.10)$$

Оскільки під час виконання курсової роботи здобувач в середньому витрачає 450 год. машинного часу, то величина фонду часу ПК дорівнює

$$T_{\text{ПК}} = 1892 * (450 / 2000) = 425.7 \text{ год} \quad (1.11)$$

Поточні витрати на експлуатацію:

Балансована вартість ПК, де  $C_p$  - ринкова вартість ПК, орієнтовно складає 40000 грн,  $k_{\text{УН}}$  - коефіцієнт, що враховує витрати на установку ПК.  $k_{\text{УН}} = 0,12$

$$C_{\text{ПК}} = C_p * (1 + k_{\text{УН}}) = 40000 * (1 + 0.12) = 44800 \quad (1.12)$$

Амортизаційні відрахування використання ПК,  $Z_{\text{АМ}}$ , обчислюються за формулою 1.13:

$$Z_{\text{АМ}} = \frac{C_{\text{ПК}}}{N_A} = \frac{44800}{5} = 8960 \quad (1.13)$$

Витрати на електроенергію ( $Z_{\text{ЕЛ}}$ ), споживану ПК, обчислюються за формулою (1.14):

$$Z_{\text{ЕЛ}} = P_{\text{ПК}} * T_{\text{ПК}} * C_{\text{ЕЛ}} * A \quad (1.14)$$

де потужність ПК,  $P_{\text{ПК}} = 0.5$  кВт; фонд корисного часу роботи ПК,  $T_{\text{ПК}} = 435.16$  год, вартість 1 кВт електроенергії для підприємств,  $C_{\text{ЕЛ}} = 1,86$  грн/кВт, коефіцієнт інтенсивного використання ПК,  $A = 0.9$ .

$$Z_{\text{ЕЛ}} = 0,5 * 435.16 * 1.68 * 0.9 = 329 \text{ грн}$$

Витрати на поточний ремонт і технічне обслуговування ПК ( $Z_p$ ) визначаються як 6% від балансової вартості ПК,  $C_{\text{ПК}}$  (формула 1.15):

$$Z_p = C_{\text{ПК}} * 0.06 \quad (1.15)$$

$$Z_p = 44800 * 0.06 = 2688 \text{ грн}$$

Непрямі витрати, пов'язані з експлуатацією ПК, визначаються як 5% від балансової вартості ПК  $C_{ПК}$  (формула 1.16):

$$Z_{MAT} = C_{ПК} * 0.05 \quad (1.16)$$

$$Z_{MAT} = 44800 * 0.05 = 2240 \text{ грн}$$

Поточні витрати на експлуатацію  $V$  " розраховуються за формулою (1.17):

$$V^{\wedge}_1 = Z_{оп} + Z_{ам} + Z_{ел} + Z_p + Z_{mat} \quad (1.17)$$

Заробітна плата обслуговуючого персоналу складає в середньому - 10000

Тож, поточні витрати на експлуатацію,  $V_1''$ , грн, складають:

$$V_1'' = 10000 + 8960 + 329 + 2688 + 2240 = 24\,217 \text{ грн}$$

А, загальні витрати на розробку програмного забезпечення комп'ютерної системи складуть:

$$V_1 = V_1' + V_1'' = 375000 + 24\,217 = 399\,217 \text{ грн} \quad (1.18)$$

Витрати на придбання і установку ПК:

$$V_2 = C_{ПК} = 44800$$

### 1.8.3. Витрати на підготовку приміщення і навчання персоналу

Витрати на підготовку приміщення  $V_3 = 0$ , так як приміщення є в наявності.

Витрати на навчання персоналу  $V_4$ .

В середньому навчання персоналу триватиме 1 місяць, тому можна вважати, що  $V_4 = 4500$  грн;

### 1.8.4. Загальна вартість розробки і впровадження системи

Загальна вартість розробки і впровадження системи вираховується за формулою (1.19):

$$V_{\Sigma} = V_1 + V_2 + V_3 + V_4 \quad (1.19)$$

$$V_{\Sigma} = 399\,217 + 0 + 0 + 4500 = 403\,717 \text{ грн}$$

Оскільки норма амортизаційних втрат для комп'ютерних систем  $HA = 5$ , то для обрахування річного економічного ефекту слід брати до розгляду величину (формула 1.20):

$$V_p = \frac{V_{\Sigma}}{HA} \quad (1.20)$$

$$V_p = \frac{403717}{5} = 80743 \text{ грн}$$

Термін окупності розробки визначається:

$$T_{\text{ок}} = \frac{1}{K_{\text{ЕФ}}},$$

де коефіцієнт економічної ефективності  $K_{\text{ЕФ}} = \frac{\Pi_p}{V_p}$ , де річний прибуток  $\Pi_p$  від впровадження системи буде досягнуто за рахунок прискорення роботи відділ, зменшенню кількості можливих помилок та витрат на їх виправлення і, орієнтовно, складатиме 10 000 грн на рік.

$$K_{\text{ЕФ}} = \frac{10\,000}{80743} = 0.123 \quad (1.21)$$

Отже, термін окупності ІС розраховується за формулою (1.22):

$$T_{\text{ок}} = 1/K_{\text{ЕФ}} \quad (1.22)$$

$$T_{\text{ок}} = \frac{1}{0.123} = 8.1 \text{ років}$$

### 1.9. Висновок до розділу 1

Складська інформаційна система (СІС) – це програмне забезпечення для управління операціями на складі. Вона дозволяє відстежувати запаси, керувати прийомом і відвантаженням товарів, контролювати переміщення продукції та забезпечувати точне виконання замовлень. СІС автоматизує облік і звітність, підвищуючи ефективність роботи складу та знижуючи ризики помилок.

ПрАТ «Оболонь», що займається виготовленням різних напоїв, має встановлені подібні інформаційні системи, але вони застарілі і мають низький рівень безпеки. Аналізуючи рішення даної проблеми, ми вирішили впровадити нову інформаційну систему. Нова система дозволить оптимізувати процеси управління та контролю за продукцією, спростить роботу персоналу та зменшить ризик помилок. Вона також забезпечить надійне ведення обліку, що допоможе уникнути втрат та забезпечить точну інформацію для прийняття управлінських рішень.

Нова інформаційна система окуповується за 8 років і є чудовою інвестицією. З урахуванням довгострокових переваг, таких як підвищення ефективності,

зменшення помилок та оптимізація процесів, така система не лише виправдовує витрати, але й приносить значну користь бізнесу. Тому, інвестиція в складську програму з окупністю в 8 років варта цього.

## **РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЕКТУВАННЯ**

### **2.1. Загальні положення.**

2.1.1 Найменування системи: «Інформаційна система для обліку готової продукції на складі ПрАТ Оболонь»

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

2.1.3 У випадку необхідності на наступних стадіях робіт по створенню системи окремі положення можуть уточнюватися і розвиватися.

### **2.2. Призначення та цілі створення системи**

2.2.1 Призначення системи обліку готової продукції на складі полягає в створенні ефективного механізму контролю та управління запасами продукції. Вона дозволяє підприємству точно відстежувати кількість, місцезнаходження товарів на складі. Також система призначена для зв'язку із клієнтами. Основні функції цієї системи включають в себе моніторинг запасів, прогнозування попиту, оптимізацію рівня запасів та керуванням постачанням, клієнтами і їх замовленнями.

2.2.2 Ціль створення системи для обліку готової продукції на складі полягає в покращенні управління складськими процесами та оптимізації руху продукції. Шляхом автоматизації та систематизації даних про запаси, система дозволяє підприємству підтримувати належний рівень запасів, уникати нестачі або перевитрат, а також забезпечувати якість та безперервність виробничих процесів. Таким чином, метою цієї системи є забезпечення ефективного та оптимального управління запасами, що сприяє підвищенню продуктивності, зниженню витрат та підвищенню задоволення клієнтів.

### **2.3. Характеристика об'єкта автоматизації**

Об'єктом автоматизації є діяльність складу підприємства - приміщення або територія, призначене для зберігання, обробки та розподілу товарів або матеріалів.

Головна функція складу полягає у тимчасовому зберіганні товарів перед їхнім подальшим використанням, продажем або доставкою до кінцевого споживача.

Основні характеристики складу включають в себе:

- площа та об'єм: склад займає площу 10 000 квадратних метрів, висота стелі 12 метри;
- інфраструктура та обладнання: склад обладнаний сучасними системами вантажопідйомності, автоматичними конвеєрами для руху товарів, системами відеоспостереження, системами контролю доступу та автоматичними воротами;
- умови зберігання: склад розподілений на зони з різними умовами зберігання, включаючи зони з контрольованою температурою та вологістю для зберігання харчових продуктів та фармацевтичних товарів;
- системи безпеки: для забезпечення безпеки товарів на складі встановлені високоякісні системи відеоспостереження та сигналізації, а також охоронна служба працює цілодобово;
- логістичні процеси: склад має відмінно організовані логістичні процеси, включаючи систему приймання товарів, інвентаризації, відбірки замовлень та відвантаження. Всі процеси автоматизовані і підтримуються спеціалізованим програмним забезпеченням;

## **2.4. Вимоги до системи**

### **2.4.1. Вимоги до системи в цілому**

Вимоги до системи в цілому включають ряд ключових аспектів, які визначають функціональність, продуктивність та надійність системи. Деякі з цих вимог можуть включати:

- **скальованість:** система повинна бути здатна масштабуватися зростанням обсягів даних та виконанням, щоб вона могла ефективно працювати навіть при збільшенні робочого навантаження;
- **надійність:** вимоги до надійності передбачають, що система повинна бути стійкою до відмов та забезпечувати неперервну доступність для користувачів;

- безпека: система повинна мати вбудовані заходи безпеки, які захищають дані від несанкціонованого доступу, а також забезпечують цілісність та конфіденційність інформації;
- швидкодія: вимоги до продуктивності передбачають, що система повинна працювати ефективно та оперативно, забезпечуючи швидку обробку запитів та відповідей користувачам;
- сумісність: система повинна бути сумісною з існуючими інформаційними та технічними системами, які використовуються в організації, забезпечуючи безпроблемну інтеграцію та обмін даними.

#### 2.4.1.1. Вимоги до структури і функціонування системи

Вимоги до структури і функціонування системи складського управління визначаються для забезпечення ефективності та оптимізації робочих процесів. Ось кілька ключових аспектів цих вимог:

- модульність: система повинна бути побудована з використанням модульної архітектури, що дозволяє легко розширювати та змінювати функціональність. Кожен модуль повинен виконувати конкретні завдання і мати чітко визначені інтерфейси для взаємодії з іншими модулями;
- інтеграція з іншими системами: система повинна бути здатна інтегруватися з іншими інформаційними системами в організації, такими як системи управління виробництвом, бухгалтерські системи та системи управління відносинами з клієнтами (CRM), для обміну даними та забезпечення єдиної точки доступу до інформації;
- автоматизація процесів: система повинна автоматизувати ключові процеси складського управління, включаючи приймання товарів, зберігання, відправлення та інвентаризацію. Це допоможе знизити ручну працю та підвищити ефективність операцій;
- забезпечення доступу і безпеки: система повинна мати механізми для контролю доступу до даних та функцій з різних рівнів користувачів. Крім того, вона повинна забезпечувати заходи безпеки для захисту від несанкціонованого доступу та збереження конфіденційності даних;

- звітність і аналітика: система повинна надавати можливості для створення різноманітних звітів та аналітичних звітів, які допоможуть в управлінні запасами, прогнозуванні попиту та прийнятті стратегічних рішень.

2.4.1.1.1. Система повинна мати клієнт-серверну архітектуру, що використовує єдину базу даних (БД). В складу це означає, що інформаційна система повинна мати централізовану БД, доступ до якої забезпечується через такі робочі місця: завідувач складу, комірник і адміністратор.

2.4.1.1.2. Діагностування функціонування системи на складі має передбачати виявлення відхилень від нормального процесу роботи, помилок програмного забезпечення та проблем з комп'ютерно-технічними засобами. Повідомлення про ці відхилення повинні бути доступні користувачам для оперативного втручання та вирішення проблем. Забезпечення взаємозв'язку між підсистемами повинно здійснюватися на інформаційному рівні через загальну БД та мережеві засоби.

2.4.1.1.3. Розвиток і модернізація системи повинні проводитися з урахуванням змін у виконуваних функціях та оновленням технічних та програмних засобів. Структура і технологія програмного забезпечення системи повинні бути гнучкими і спроможними розширюватися, збільшувати обсяг задач і оброблюваної інформації. Забезпечення сумісності програмного та інформаційного середовища має здійснюватися за допомогою загальносистемного протоколу обміну та єдиної системи класифікації і кодування.

2.4.1.1.4. Функціонування системи складського управління повинно забезпечувати як діалогову, так і мережну (розподілену) обробку даних для оптимальної ефективності та швидкодії. Щодо діалогової обробки, система має забезпечувати зручний інтерфейс взаємодії з користувачем, де він може вводити дані, здійснювати пошук і отримувати результати обробки даних у реальному часі. Одночасно, мережна обробка даних передбачає можливість розподіленої роботи над спільними даними через мережу, що дозволяє користувачам працювати з системою незалежно від їхнього місця знаходження. Важливо також забезпечити ефективну синхронізацію даних між різними пристроями та користувачами для забезпечення єдиної та актуальної інформації в системі.

#### 2.4.1.2. Вимоги до чисельності і кваліфікації персоналу

2.4.1.2.1. Вимоги до чисельності та кваліфікації персоналу для системи складського управління є критичним аспектом її успішної реалізації. Щодо чисельності, вона повинна бути адаптована до розміру та потреб складу. Це може включати менеджерів, операторів, технічний персонал і, можливо, працівників, відповідальних за приймання та відвантаження товарів. Однак, наряду зі збільшенням чисельності, важливо враховувати і кваліфікацію персоналу. Рівень кваліфікації повинен відповідати потребам системи, включаючи знання про складське управління, вміння працювати з інформаційними системами та обладнанням, а також здатність ефективно спілкуватися та співпрацювати з іншими членами команди. Відповідна підготовка та навички персоналу можуть позитивно вплинути на продуктивність та ефективність складських операцій, забезпечуючи надійну та ефективну роботу всієї системи управління складом.

2.4.1.2.2. На складі користувачами системи можуть бути адміністратор, лаборанти, завідувач складу та комірник.

#### 2.4.1.3. Показники призначення

2.4.1.3.1. Показники призначення для складської інформаційної системи (СІС) повинні відображати ступінь та якість автоматизації планової, інформаційно-облікової і управлінської діяльності для оптимального функціонування складу. Оцінка може включати такі показники, як ступінь автоматизації процесів управління запасами, точність та швидкість обробки інформації про приймання та відвантаження товарів, а також ефективність управління запасами через використання автоматизованих аналітичних інструментів. Для досягнення цільового призначення системи, перелік і допустимі значення цих показників повинні бути чітко визначені на стадії технічного проектування, щоб забезпечити оптимальне функціонування та відповідність потребам складського управління.

2.4.1.3.2. Система повинна мати можливість налаштування на параметри ресторану та периферійного обладнання при її модернізації та розвитку, а також зміні процесів та методів організаційного управління.

#### 2.4.1.4. Вимоги до надійності

2.4.1.4.1. Система є багатофункціональною і призначена для використання протягом робочого дня. Всі функції системи виконуються дискретно. Оцінка надійності проводиться по кожній функції окремо. Основними показниками надійності є:

- ймовірність безвідмовного виконання задачі в заданий термін ( $L_i$ );
- коефіцієнт готовності програмно-технічного комплексу ( $K_r$ );
- середній час відновлення програмно-технічного комплексу ( $T_v$ );
- мінімальний час між двома відмовами за календарний місяць ( $T_e$ ).

2.4.1.4.2. Комплекс технічних засобів повинен передбачати:

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

Для забезпечення надійності програмного та інформаційного забезпечення необхідно передбачити використання:

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

#### 2.4.1.5. Вимоги до безпеки

Для забезпечення безпеки при експлуатації, налагодженні, монтажі, обслуговуванні і ремонті технічних засобів системи потрібно дотримуватись вимог ДСТУ: ДСТУ 2293-99, ДСТУ ISO 6309:2007, ДСТУ 12.0.230:2008, ДСТУ 7237:2011, ДСТУ 7238:2011, ДСТУ 7239:2011; по доступним рівням освітленості, вібраційних і шумових навантажень слід дотримуватися вимог

відповідно ДСТУ Б А.3.2-15:2011, ДСТУ EN 14253:2018, ДСТУ 2867-94.

#### 2.4.1.6. Вимоги з ергономіки та технічної естетики

Вимоги з ергономіки та технічної естетики системи повинні відповідати стандартам ДСТУ, зокрема, ДСТУ 8604:2015 і ДСТУ 7298:2013. Ці стандарти містять норми щодо організації робочого простору, дизайну інтерфейсів користувача, розміщення обладнання та освітлення. Наприклад, ергономічні вимоги можуть включати оптимальні розміри та форму елементів інтерфейсу, зручність розміщення та доступ до елементів управління, а також забезпечення комфортних умов праці для користувачів. З естетичної точки зору, вимоги можуть стосуватися зовнішнього вигляду системи, використання кольорів, матеріалів та інших аспектів дизайну, щоб забезпечити гармонійний і привабливий зовнішній вигляд. Освітленість також є важливим аспектом ергономіки, і вона може бути регульована з урахуванням вимог безпеки та комфорту користувачів.

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

2.4.1.7.1. Система повинна бути обслуговувана згідно з вимогами, визначеними в стандартах. Загальні вимоги щодо експлуатації, технічного обслуговування і ремонту повинні відповідати стандарту ДСТУ 3576-97.

2.4.1.7.2. Розміщення технічних засобів системи повинно відповідати вимогам, визначеним у будівельних нормах і правилах (ДБН В.2.2-9-2009). Напруга живлення системи повинна бути 220/380 В змінного струму і частотою  $(50 \pm 1)$  Гц. Допустиме відхилення напруги від +10% до -15%, а тривалість перерв у живленні не повинна перевищувати 0,001 с.

2.4.1.7.3. Кількість, кваліфікація і режим роботи персоналу, який обслуговуватиме систему, повинні відповідати вимогам, зазначеним у технічних умовах і інструкціях з експлуатації.

2.4.1.7.4. Склад, розміщення і умови зберігання компонентів системи повинні відповідати рекомендаціям, зазначеним у документації на ці компоненти.

2.4.1.7.5. Регламент обслуговування повинен відповідати рівню і умовам роботи системи, щоб у разі відмови забезпечити аварійний режим роботи.

#### 2.4.1.8. Вимоги до захисту інформації від несанкціонованого доступу:

- Операційні системи сервера, які мають відповідні засоби захисту;
- Локальна мережа та програми захисту в мережі, такі як брандмауери (firewalls);
- клієнт-серверна система управління базами даних (СУБД) з можливістю використання тригерів, представлень, процедур, функцій та встановлення груп користувачів і ролей доступу;
- аутентифікація і авторизація: система повинна мати механізми аутентифікації, які підтверджують ідентичність користувачів, а також механізми авторизації, що контролюють доступ користувачів до різних ресурсів системи;
- фізичний захист: забезпечення фізичної безпеки серверних пристроїв та інших компонентів системи для запобігання несанкціонованого доступу до обладнання;
- захист від зовнішніх загроз: застосування заходів захисту від вірусів, шкідливих програм та інших зовнішніх загроз, що можуть використовуватися для отримання несанкціонованого доступу до системи.

#### 2.4.1.9. Вимоги щодо збереження інформації при аваріях

2.4.1.9.1. Система повинна мати механізми автоматичного резервного копіювання даних для періодичного створення резервних копій інформації. Це дозволить відновити дані у випадку втрати або пошкодження основних даних під час аварії.

#### 2.4.1.9.2. Резервний архів і БД повинні знаходитись на різних носіях чи Пристроях

#### 2.4.1.10. Вимоги по захисту від впливу зовнішніх дій

2.4.1.10.1. Електрична складова електромагнітного поля завод не повинна перевищувати 0,3 В/м<sup>2</sup> в діапазоні частот від 0,15 до 300 МГц. Для захисту від електромагнітних полів та індустриальних завод слід використовувати екрани та фільтри.

2.4.1.10.2. Засоби, які виключають вплив шкідливих факторів на функціонування системи, повинні бути запроектовані згідно з будівельними нормами і правилами (ДБН В.2.2-9-2009). Обчислювальні засоби повинні відповідати стандартам стійкості до зовнішніх впливів (ДСТУ 2506-94)

2.4.1.11. Вимоги до патентної чистоти

При створенні даної системи не проводилися патентні дослідження.

2.4.1.12. Вимоги щодо стандартизації і уніфікації

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

2.4.2. Вимоги до функцій інформаційної системи для обліку готової продукції на складі ПрАТ «Оболонь»

2.4.2.1. Перелік функцій із зазначенням вхідної та вихідної інформації наведено в таблиці 2.1

Таблиця 2.1. Перелік функцій, вхідної та вихідної інформації

<b>№ п/п</b>	<b>Найменування функції</b>	<b>Вхідна інформація</b>	<b>Вихідна інформація</b>
1	Формування та виведення запасів продукції	Таблиці БД «Продукція», «Категорії», «Зона_зберігання», «Склад»	Форма зі списком продукції із певною категорією, складом і зоною зберігання
2	Формування та виведення зони зберігання продукції	Таблиці БД «Зона_зберігання»	Форма зі списком зон зберігання
3	Формування та виведення категорії продукції	Таблиці БД «Категорія»	Форма зі списком категорій
4	Формування та виведення інформації про склади підприємства	Таблиці БД «Склад»	Форма зі списком складів

Продовження таблиці 2.1. Перелік функцій, вхідної та вихідної інформації

№ п/п	Найменування функції	Вхідна інформація	Вихідна інформація
5	Формування та виведення даних про замовників	Таблиці БД «Замовник»	Форма зі списком замовників
6	Формування та виведення переліку автотранспорту для перевезення продукції	Таблиці БД «Автотранспорт»	Форма зі списком автотранспорту
7	Формування та виведення інформації про водіїв	Таблиці БД «Водій»	Форма зі списком водіїв
8	Формування та виведення інформації про надходження продукції	Таблиці БД «Ордер_надходження», «Склад», «Продукція», «Зона зберігання»	Форма зі списком надходжень продукції до певного складу, зони зберігання
9	Формування та виведення інформації про продаж продукції	Таблиці БД «Ордер_вибуття», «Склад», «Продукція», «Замовник»	Форма зі списком продажу
10	Формування та виведення списання продукції	Таблиці БД «Акт_списання», «Склад», «Продукція»	Форма зі списком списання продукції зі складів
11	Формування та виведення ТТН	Таблиці БД «ТТН», «Склад», «Продукція», «Замовник», «Водій», «Транспорт»	Форма зі списком ТТН
12	Пошук інформації про продукцію	Запит користувача, «Продукція»	Форма з результатами пошуку
13	Пошук інформації про категорії продукції?	Запит користувача, «Категорія»	Форма з результатами пошуку

## Продовження таблиці 2.1. Перелік функцій, вхідної та вихідної інформації

№ п/п	Найменування функції	Вхідна інформація	Вихідна інформація
14	Пошук інформації про зони зберігання продукції	Запит користувача, «Зона_зберігання»	Форма з результатами пошуку
15	Пошук інформації про замовників	Запит користувача, «Замовник»	Форма з результатами пошуку
16	Пошук інформації про автотранспорт	Запит користувача, «Автотранспорт»	Форма з результатами пошуку
17	Пошук інформації про водіїв	Запит користувача, «Водій»	Форма з результатами пошуку
18	Пошук інформації про надходження продукції	Запит користувача, «Ордер_надходження»	Форма з результатами пошуку
19	Пошук інформації про продажі продукції	Запит користувача, «Ордер_вибуття»	Форма з результатами пошуку
20	Пошук інформації про списання продукції	Запит користувача, «Акт_списання»	Форма з результатами пошуку
21	Пошук інформації про ТТН	Запит користувача, «ТТН»	Форма з результатами пошуку
22	Формування документів для друку	Таблиці БД: «ТТН», «Продукція», «Замовник», «Водій», «Транспорт», «Зона_зберігання», «Категорія», «Ордер_надходження», «Ордер_вибуття», «Акт_списання»	PDF-документ із відповідними даними

2.4.3. Вимоги до видів забезпечення для інформаційної системи обліку готової продукції на складі ПрАТ «Оболонь»

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

2.4.3.2. Вимоги до інформаційного забезпечення (ІЗ)

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

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

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

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

2.4.3.3. Вимоги до лінгвістичного забезпечення (ЛЗ)

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

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

2.4.3.4. Вимоги до програмного забезпечення (ПЗ)

2.4.3.4.1. Загальносистемне ПЗ має забезпечувати надійне і якісне виконання функціональних завдань системи. До загальносистемного ПЗ належить:

- будь-яка операційна система;
- система управління БД PostgreSQL.

2.4.3.4.2. Загальні вимоги до системного ПЗ можна сформулювати так:

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

2.4.3.4.3. Вимоги до ОС:

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

2.4.3.4.4. Вимоги до СУБД:

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

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

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

2.4.3.4.6. При розробленні спеціального ПЗ слід виконати наступні вимоги:

- використані програми мають бути сумісні між собою та із загальносистемним ПЗ;
- ПЗ має розроблятися засобами об'єктно-орієнтованого програмування;
- забезпечити відповідність інтерфейсу користувача стандартам Windows;
- необхідна модульна структура програм;
- повинна бути передбачена можливість розширення складу задач у

відповідності з новими функціональними потребами;

- ПЗ не повинно залежати від типу зовнішніх пристроїв (принтерів, дисків, сканерів тощо);
- діалог із користувачем повинен проводитись за допомогою клавіатури або миші з поясненням виконання дій і можливістю отримання підказки.

#### 2.4.3.5. Вимоги до технічного забезпечення

2.4.3.5.1. Технічні засоби системи (таблиця 2.2) повинні забезпечувати виконання функцій, перерахованих в таблиці 2.1.

2.4.3.5.2. Засоби обчислювальної техніки повинні забезпечувати обмін інформації в об'ємах, приведених в п. 2.4.3.2.

Таблиця 2.2. Вимоги до технічного забезпечення системи

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

#### 2.4.3.6. Вимоги до технічного забезпечення

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

#### 2.4.3.7. Вимоги до організаційного забезпечення

2.4.3.7.1. Організаційне забезпечення системи розробляється в відповідності з вимогами державного стандарту по АСУП.

2.4.3.7.2. При впровадженні системи не передбачається збільшення штатної чисельності підприємства. Територіальне розміщення робочих місць, на яких буде встановлена система, визначається підприємством.

#### 2.4.3.7.3. До функціонування системи висуваються наступні вимоги:

- наказом директора визначається список співробітників, які мають доступ до системи;
- контроль і прийняття рішень при аварійних ситуаціях при експлуатації

системи здійснює відповідальний за систему.

## **2.5. Склад і зміст робіт по створенню системи**

2.5.1. Стадії створення системи і терміни виконання робіт наведені в таблиці

Таблиця 2.3. Найменування робіт при створенні системи

<b>№ п/п</b>	<b>Найменування робіт</b>	<b>Строки виконання робіт</b>
1	Передпроектне дослідження об'єкта автоматизації	20.10.2023
2	Технічне завдання	15.12.2023
3	Технічний проєкт	10.01.2024
4	Оформлення документації	10.05.2024

## **2.6. Порядок контролю і приймання системи для інформаційної системи обліку готової продукції на складі ПрАТ «Оболонь»**

2.6.1. Введення в дію системи на діючому ДП СТХТ НУХТ

При введенні в дію система повинна пройти приймальні випробування відповідно до вимог ДСТУ 3974-2000.

2.6.2. Випробування працездатності та рішення про приймання в дослідну експлуатацію

Розробники разом із замовником проводять випробування для визначення працездатності системи.

Програму випробувань складає розробник і затверджує замовник.

2.6.3. Дослідна експлуатація та формування переліку доробок

Здача в дослідну експлуатацію здійснюється на основі технічного завдання та інструкції користувача.

Після дослідної експлуатації формується перелік доробок, які потребують виконання, та рекомендовані строки їх виконання.

2.6.4. Акт здачі-прийому

Введення в дію системи оформлюється актом здачі-прийому, який підтверджує передачу системи з розробників належним чином забезпеченому замовнику.

## **2.7. Вимоги до складу і змісту робіт із підготовки до введення системи в дію для інформаційної системи обліку готової продукції на складі ПрАТ «Оболонь»**

### **2.7.1. Укомплектування технічних засобів**

Замовник забезпечує укомплектування необхідними технічними засобами для роботи системи.

### **2.7.2. Навчання користувачів системи**

Замовник організовує навчання користувачів системи щодо роботи на персональних комп'ютерах та ознайомлення з інструкцією щодо експлуатації системи.

### **2.7.3. Дослідна експлуатація і введення системи в дію**

Замовник проводить дослідну експлуатацію системи з метою перевірки її працездатності та визначення можливості введення в експлуатацію.

## **2.8. Вимоги до документації для інформаційної системи обліку готової продукції на складі ПрАТ «Оболонь»**

### **2.8.1. Склад документації:**

Для системи розробляється комплекс документації, що включає технічне завдання та технічний проект.

### **2.8.2. Відповідність стандартам**

Документація на систему розробляється відповідно до вимог, до державних стандартів серії 19 "Єдина система програмної документації" та серії 24 "Єдина система стандартів автоматизованих систем управління".

## **2.9. Джерела розробки для інформаційної системи обліку готової продукції на складі ПрАТ «Оболонь»**

### **2.9.1. Використання документів**

При розробленні технічного завдання на систему використано такі документи:

- ДСТУ 3008-2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлення;

- ДСТУ 3973–2000. Система розроблення та поставлення продукції на виробництво.;
- ДСТУ Б В.2.5–82:2016. Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом.

## РОЗДІЛ 3. ОПИС КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ

### 3.1. Інформаційне забезпечення системи

Логічна модель, що наведена в додатку В інформаційної системи обліку готової продукції на складі ПрАТ «Оболонь» включає в себе сутності з детальним описом їх структури та взаємозв'язків.

Сутність "Продукція":

- код\_продукції (первинний ключ);
- назва\_продукції;
- кількість\_палетів;
- ціна\_за\_од (ціна продукцію за одиницю);
- код\_зони\_зберігання (зовнішній ключ, що посилається на таблицю «Зона зберігання»);
- код\_складу (зовнішній ключ, що посилається на таблицю «Склад»);
- артикул;
- код\_категорії (зовнішній ключ, що посилається на таблицю «Категорія»).

Сутність "Зона\_зберігання":

- код\_зони\_зберігання(первинний ключ) ;
- назва\_зони\_зберігання;
- місткість\_палетів ;
- температура\_зберігання ;
- код складу (зовнішній ключ, що посилається на таблицю "Склад") .

Сутність "Категорія":

- код категорії (первинний ключ);
- назва категорії;
- код складу (зовнішній ключ, що посилається на таблицю "Склад") .

Сутність "Склад":

- код складу (первинний ключ);
- назва складу.

Сутність "Замовник":

- код\_замовника(первинний ключ);
- ПІБ\_замовника (назва типу страви);
- телефон\_замовника;
- статус\_замовника (активний чи ні);
- e\_mail (пошта);
- назва\_підприємства;
- адреса;
- Реквізити\_розрахунковий\_рахунок\_.

Сутність "Співробітник":

- код\_співробітника (первинний ключ);
- ПІБ\_співробітника;
- пароль;
- e\_mail.

Сутність "Ордер\_надходження":

- код\_ордеру\_надходження (первинний ключ);
- номер\_ордеру\_надходження;
- код\_складу (зовнішній ключ, що посилається на таблицю «Склад»);
- код\_продукції (зовнішній ключ, що посилається на таблицю «Продукція»);
- код\_зони\_зберігання (зовнішній ключ, що посилається на таблицю «Зона\_зберігання»);
- кількість\_палетів\_;
- номер\_партії;
- цех;
- сума;
- дата.

Сутність "Ордер\_вибуття":

- код\_ордеру\_вибуття (первинний ключ);

- номер\_ордеру\_вибуття;
- код\_замовника (зовнішній ключ, що посилається на таблицю «Замовник»);
- код\_складу (зовнішній ключ, що посилається на таблицю «Склад»);
- код\_продукції (зовнішній ключ, що посилається на таблицю «Продукція»);
- код\_зони\_зберігання (зовнішній ключ, що посилається на таблицю «Зона\_зберігання»);
- кількість\_палетів\_;
- сума;
- дата.

Сутність "Акт\_списання ":

- код\_акту\_списання (первинний ключ);
- номер\_акту\_списання;
- код\_складу (зовнішній ключ, що посилається на таблицю «Склад»);
- код\_продукції (зовнішній ключ, що посилається на таблицю «Продукція»);
- причина\_списання
- кількість\_палетів\_;
- дата.

Сутність "Водій":

- код\_водія (первинний ключ);
- ПІБ\_водія;
- телефон\_водія.

Сутність "Автотранспорт":

- код\_транспорту (первинний ключ);
- назва\_транспорту;
- номер\_транспорту;
- вантажопідйомність.

Сутність "ТТН":

- код\_ТТН (первинний ключ);
- номер\_ТТН;
- код\_замовника (зовнішній ключ, що посилається на таблицю «Замовник»);
- код\_складу (зовнішній ключ, що посилається на таблицю «Склад»);
- код\_продукції (зовнішній ключ, що посилається на таблицю «Продукція»);
- кількість\_палетів\_;
- ціна\_за\_од;
- сума;
- код\_транспорту (зовнішній ключ, що посилається на таблицю «Автотранспорт»);
- код\_водія (зовнішній ключ, що посилається на таблицю «Водій»);
- дата.

Фізична модель, що наведена в додатку Ж, включає структуру таблиць, атрибути, типи даних та індекси. Вона залежить від конкретної системи керування базами даних (СКБД), яка буде використовуватися.

Структура бази даних:

- таблиця "Продукція" зі зазначеними атрибутами і відповідними типами даних;
- таблиця "Зона\_зберігання" з атрибутами і типами даних;
- таблиця "Склад " з атрибутами і типами даних;
- таблиця "Категорія" з атрибутами і типами даних;
- таблиця "Замовник" з атрибутами і типами даних;
- таблиця "Ордер\_надходження" з атрибутами і типами даних;
- таблиця " Ордер\_вибуття" з атрибутами і типами даних;
- таблиця "Акт\_списання" з атрибутами і типами даних;
- таблиця "Водій " з атрибутами і типами даних;

- таблиця "Автотранспорт " з атрибутами і типами даних;
- таблиця "ТТН " з атрибутами і типами даних.

Кожна таблиця має атрибути із обов'язковим первинним ключем. Може бути зовнішній ключ, який посилається на іншу таблицю, а також можуть бути встановлені індекси для покращення швидкості пошуку та фільтрації даних.

### 3.2. Алгоритмізація та реалізація комплексу задач автоматизації

Перший кроком для існування інформаційної системи для обліку готової продукції на складі ПрАТ «Оболонь» ми розробили бізнес-модель роботи ІС, яка наведена в додатку Б (0). Наступним кроком було створення логічної і фізичної моделі системи за допомогою AllFusion ERWin Data Modeler. Далі ми згенерували нашу базу даних в AllFusion ERWin Data Modeler у вигляді sql-запиту і виконали його в СКБД PostgreSQL server.

Для реалізації нашої ІС використано графічну бібліотеку на мові Python під назвою tkinter. Це стандартний інтерфейс для створення графічного користувацького інтерфейсу. Він базується на бібліотеці Tk. Tkinter постачається разом зі стандартною бібліотекою Python і дозволяє розробникам легко створювати вікна, кнопки, поле введення, меню та інші елементи інтерфейсу користувача.

Для з'єднання із базою даних і виконання запитів використано спеціальну бібліотеку для з'єднання із PostgreSQL під назвою Psycopg2.

Як працює ця бібліотека і як виконуються запити розглянуто в додатку Е.

У коді представлено імпорт бібліотеки psycopg2, метод на додавання даних `execute_sql_query_insert`, метод на отримання даних `execute_sql_query_get` і саме підключення до бази даних, що зберігається в змінній `conn_str`.

Перейдемо до створення графічних об'єктів. Бібліотека tkinter має в собі клас TK(). Створюючи екземпляр класу ми створюємо вікно програми.

Наведемо приклад з розробленої ІС:

```
root = Tk()
root.title('Вхід')
root.mainloop()
```

`root` – це екземпляр класу `Tk()`, який приймає всі функції цього класу. `title()` – це функція, за допомогою якої можна задавати назву вашого вікна. `mainloop()` – функція, яка відображає ваше вікно, без неї вікно не відкриється.

Для створення текстового напису є функція `Label`, яка має в собі іменовані аргументи `master` - який вказує на батьківський віджет, до якого потрібно додати новий віджет. Є аргумент `text`, за допомогою якого можна встановлювати текст вашому віджету, `bg (background)` за допомогою якого можна встановлювати колір заднього фону, `font` – для встановлення шрифту тексту тощо.

Наведемо приклад створення текстового напису:

```
descr = Label(root, text='Вхід', font=('Arial', 18, 'bold'), background='White')
descr.place(x=432, y=20)
```

Зміна `descr`, якій присвоєно функцію `Label`, який встановлений на нашому створеному вікні `root` (аргументу `master` присвоєно віджет `root`) із текстом «Вхід», із жирним шрифтом `Arial` 18 розміру та білим заднім фоном. Функція `place` розміщує віджет на вікні із заданими координатами `x` і `y`.

Так само можна створити і розмістити будь-який графічний елемент. Розглянемо як працювати із графічним елементом `Button` (кнопка):

```
b_entr = Button(self.root, width=30, text='Увійти', command=go_log)
b_entr.place(x=350, y=220)
```

Все подібно до елементу текстового напису, але головна відмінність це те, що кнопка повинна виконувати якусь дію. Ця дія прописана в функції `go_log` (назва функції, яка містить свою логіку виконання), її присвоюємо до аргументу `command`. Подивитись невеликий приклад створення вікна додавання складу нашої інформаційної системи можна в додатку Е:

Функція `entry_data`, з додатку, створює нове графічне вікно, яке наслідується від батьківського функцією `Toplevel`(назва вашого вікна у вигляді змінної). Створено текстовий напис `label` «Введіть назву складу», поле вводу назви складу `entry_skl` і кнопку «Додати» `button_add`, що викликає внутрішню функцію `add()` для обробки і додавання даних до таблиці «Склад»

Результат створення форми продемонстровано на рисунку 3.1

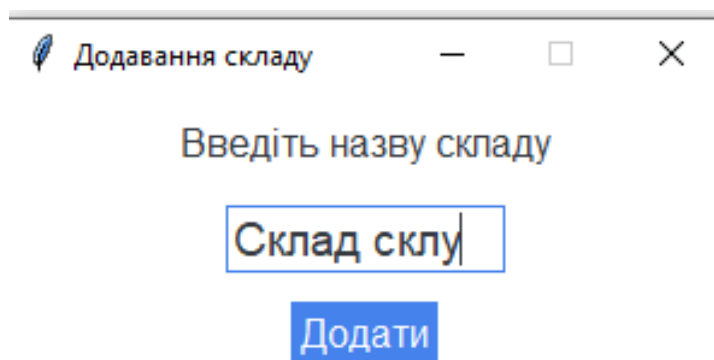


Рисунок 3.1 - Створене вікно додавання складу

Таким чином можна створювати графічні елементи за допомогою даної графічної бібліотеки.

Загалом, для даної інформаційної системи створені такі вікна:

1. Вікно авторизації
2. Вікно реєстрації
3. Вікно вибору складу, з яким користувач хоче працювати
4. Основне вікно, де розташовані складські функції

Вхід автоматизовано таким чином:

1. Створено поля, куди користувач вводить свої дані для входу.
2. Натискаючи на кнопку «Увійти» виконується блок коду перевірки інформації:

```
sql_query = "SELECT * FROM Співробітник WHERE ПІБ_співробітника = %s AND Пароль = %s;"
```

```
params = (name, password)
```

```
res = execute_sql_query_get(conn_str, sql_query, params)
```

```
if res and len(res) > 0:
```

```
    self.open_form3()
```

```
else:
```

```
    self.data_wrong.place(x=462, y=185)
```

res – результуюча змінна, куди потрапляє результат виконання запиту execute\_sql\_query\_get. Якщо він не пустий, то відкривається форма self.open\_form3() – форма вибору складу.



Рисунок 3.2 - Вікно авторизації

Реєстрація автоматизована таким же чином як і авторизація, тільки дані, які вводить користувач потрапляють до бази даних. Виконується sql-функція insert.

```
if psw2 != psw and len(psw)>0 and len(psw2)>0:
```

```
    error_saving_psw(self.new_window)
```

```
    return
```

```
if not empty_fields:
```

```
    for field_name in self.error_labels.keys():
```

```
        if field_name not in empty_fields:
```

```
            self.error_labels[field_name].destroy()
```

```
            sql_query = f"INSERT INTO Співробітник (ПІБ_співробітника, Пароль, е_mail)
```

```
VALUES (N'{name} {lastname}" \
```

```
f" {father_name}', N'{psw}',N'{email}');"

```

```
execute_sql_query_insert(conn_str, sql_query)
```

```
self.open_login()
```

```
success_login(self.new_window)
```

psw2 і psw – це поля вводу двох паролей. Якщо вони співпадають і не має пустих полів, то виконується запит на додавання даних.

Рисунок 3.3 - Вікно реєстрації

Вікно авторизації та реєстрації має можливість скривати і відкривати пароль.

Натискаючи на кнопку «око» викликається така функція:

```
def toggle_password(self):
    self.show_password = not self.show_password
    if self.show_password:
        self.entry2.config(show='*')
        self.toggle_button.config(image=self.hide_image,          compound='right',
bd=0, relief=FLAT)
    else:
        self.entry2.config(show='')
        self.toggle_button.config(image=self.show_image,          compound='right',
bd=0, relief=FLAT)
```

Завантажено два зображення: «відкрите око» і «закрите око» (приклад завантаження зображення продемонстровано нижче), якщо скриваємо пароль, то поле `entry2` демонструє нам замість тексту зірочки.

При виборі складу на формі `self.open_form3()`, виконується функція `choose_manu(root, self)`, де `root` – батьківська форма, а `self` – об'єкт, за допомогою якого звертаємось до елементів класу батьківського вікна.

Функція заносить обраний склад з назвою до змінної `self.chosen_name` і відкривається обраний склад

Головне меню формувалось створенням свого фрейму головного меню в конструкторі класу та, переважно, створенням кнопок, що відкривають свій `frame`

Код створення фрейму до головного меню:

```
self.frame_menu = tk.Frame(self.new_window, bg="White")
self.frame_menu.pack(fill='both', expand=True)
```

`self.new_window` відноситься до аргумента `master` – вікна, на якому буде встановлено графічний віджет, а `fill` і `expand`. `fill` вказує, як розтягнути віджет в батьківському контейнері. У даному випадку, `fill='both'` означає, що віджет повинен розтягнутись і по горизонталі, і по вертикалі, щоб він повністю заповнив доступне простору в батьківському контейнері. `expand` вказує, чи може віджет розтягуватись, щоб заповнити весь вільний простір в батьківському контейнері, якщо такий простір є. У випадку `expand=True`, віджет буде розтягуватись, якщо в батьківському контейнері є вільне місце.

Код конструктора класу із усіма створеними фреймами і додатковими графічними віджетами продемонстровано в додатку E

Виклик головного меню:

```
def menu(self,chosen_name)
```

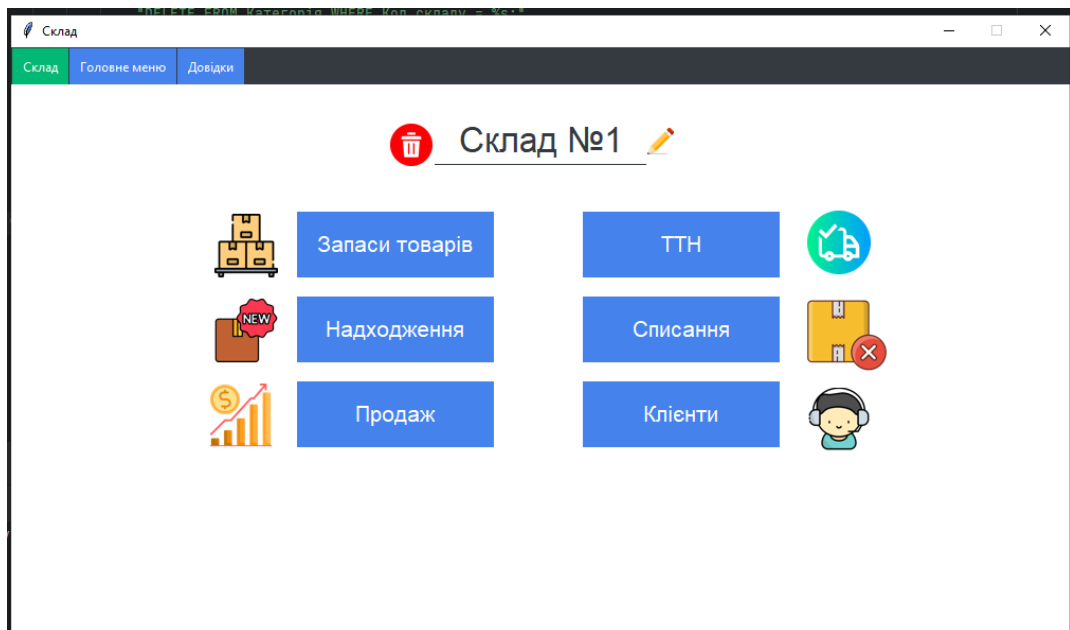


Рисунок 3.4 - Головне меню системи

Цікаво як відобразити зображення на формі. Для цього треба спочатку його відкрити за шляхом його розміщення в папках і настройкою розміру.

Код відкриття зображення, зміни його розміру зі збереженням в змінну:

```
original_image_prod = Image.open('./images/free-icon-product-3712193.png')
resized_image_prod = original_image_prod.resize((60, 60))
self.prod = ImageTk.PhotoImage(resized_image_prod)
```

Для цього знадобилась бібліотека PIL, яка призначена для роботи з зображеннями. Вона надає можливості для завантаження, збереження, зміни розміру, обрізки, обробки та маніпулювання зображеннями різних форматів.

Додавання зображення на текстовий напис label1 до аргументу image:

```
label1 = tk.Label(self.frame_menu, image=self.prod, bg="White")
label1.place(x=190, y=120)
```



Рисунок 3.5 - Приклад завантажених зображень на текстових написах label

Розглянемо каскадне меню, яке викликаємо натисканням на «Довідки»:

```
def dovidku(self):
    menu = tk.Menu(self.frame, tearoff=0)
    menu.add_command(label="Зони зберігання", compound="right",
        command=self.open_save_menu)
    menu.add_command(label="Категорії", command=self.open_category_menu)
    menu.add_command(label="Водії",
        compound="right",command=self.open_driver_menu)
    menu.add_command(label="Авто",
        compound="right",command=self.open_avto_menu)
    menu.configure(bg="white", fg="black", activebackground="cornflower blue",
        activeforeground="black")
    x = self.new_window.winfo_rootx()
    y = self.new_window.winfo_rooty()
    menu.post(x + 156, y + 35)
```

Натискаючи на пункти з меню довідок, відкривається фрейм з відповідними обраними даними.

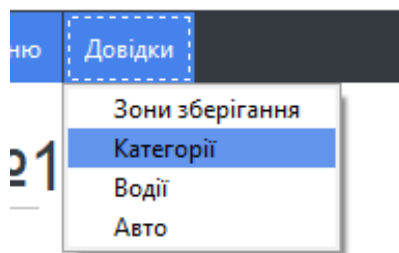


Рисунок 3.6 - Створене каскадне меню для довідок

Для видалення поточного складу виконується такий код:

```
def del_c_scl():
    sql_query = f"""\SELECT Код_складу FROM Склад WHERE Назва_складу
= %s"""""
    params = (self.chosen_name,)
    scl_id = execute_sql_query_get(conn_str, sql_query, params)
    print(scl_id[0][0])

sql_queries = [
    "DELETE FROM Акт_списання WHERE Код_складу = %s;",
    "DELETE FROM ТТН WHERE Код_складу = %s;",
    "DELETE FROM Ордер_вибуття WHERE Код_складу = %s;",
    "DELETE FROM Ордер_надходження WHERE Код_складу = %s;",
    "DELETE FROM Продукція WHERE Код_складу = %s;",
    "DELETE FROM Категорія WHERE Код_складу = %s;",
    "DELETE FROM Зона_зберігання WHERE Код_складу = %s;",
    "DELETE FROM Склад WHERE Код_складу = %s;"
]
# Виконання SQL-запитів для очищення таблиць
for sql_query in sql_queries:
    params = (scl_id[0][0],)
    execute_sql_query_insert(conn_str, sql_query, params)
if len(get_scl())>=1:
```

```
self.menu(get_scl()[0][-1])
```

else:

```
self.frame_menu.pack_forget()
```

```
self.create_empty_frame()
```

Видаляються всі поточні записи із даним складом, а потім видаляється й сам склад.

Подібні Кнопки такі як: запаси товарів, надходження, продаж, скривають фрейм головного меню і відкривають свій frame із відповідними даними.

Якщо натиснути на кнопку «Запаси товарів», то скриється фрейм головного меню і відкриється фрейм для запасів товару.

Код відкриття фрейму «Запаси товарів»:

```
self.frame_menu.pack_forget()
```

```
for widget in self.zapas_menu.winfo_children():
```

```
widget.destroy()
```

```
self.zapas_menu.pack(fill='both', expand=True)
```

```
self.zap_name = tk.Label(self.zapas_menu, bg="White", bd=0, text="Запаси товарів",  
font=("Arial", 25))
```

```
self.zap_name.pack(pady=20)
```

№	Назва продукції	Кількість палетів	Ціна за од	Зона зберігання	Склад	Артикул	Категорія
810	халва	2	3	їжа	Склад №1	4TRHL	алкоголка
773	Барбарис	0	5	їжа	Склад №1	R1NIC	категорія А
876	efef2	0	3	зона б1	Склад №1	1PM7F	алкоголка
774	Лимон	2	22	холоднаа	Склад №1	J1WTO	категорія А
771	Хліб	1	32	нова зона	Склад №1	41097	незнаю
770	сало	14	22	їжа	Склад №1	C6LKC	папалам
844	уауа34	9	3	зона б1	Склад №1	1WJ8Z	алкоголка
737	уауа	241	3	зона б1	Склад №1	84433	папалам
772	Абрикос	51	10	їжа	Склад №1	CK0QC	фрукт
775	масло	54	23	холоднаа	Склад №1	L0AD4	папалам
776	кабан	2	3	зона б1	Склад №1	YUI0X	алкоголка

Рисунок 3.7 - Фрейм «Запаси товарів»

На фреймі всі кнопки виконують sql-запит. Створено таблицю, яка демонструє нам дані про продукцію за допомогою Treeview - це віджет, який надає інтерфейс для відображення і обробки ієрархічних даних у вигляді таблиці з деревом.

Код створення такої таблиці:

```
columns = ("№", "Назва продукції", "Кількість палетів", "Ціна за од",
           "Зона зберігання", "Склад", "Артикул", "Категорія")
tree = ttk.Treeview(self.zapas_menu, columns=columns, show="headings")
```

ttk – це скорочення ttkbootstrap при імпорті:

```
import ttkbootstrap as ttk
```

ttkbootstrap – це теж саме, що і tkinter, але більш модернізована бібліотека з новими оновленими графічними елементами.

При відкриванні фрейму «Запаси товарів» виконується одразу ж запит, який зчитує всі дані таблиці «Продукція»:

```
sql_get_prod = f"""
SELECT p.Код_продукції, Назва_продукції, Кількість_палетів_, Ціна_за_од,
z."Назва_зони_зберігання",
Назва_складу, Артикул, Категорія
FROM Продукція p
LEFT JOIN "Зона_зберігання" z ON p."Код_зони_зберігання" =
z."Код_зони_зберігання"
LEFT JOIN "Склад" s ON p."Код_складу" = s."Код_складу"
LEFT JOIN "Категорія" k ON p."Код_категорії" = k."Код_категорії"
WHERE p."Код_складу" = {self.current_sclad_id};
"""

data = None
while data is None:
data = execute_sql_query_get(conn_str, sql_get_prod)
populate_tree(tree, data)
```

результуючу змінну додаємо до функції додавання даних до нашої таблиці Treeview.

Функція додавання:

```
def populate_tree(tree, data):
    for row in data:
        tree.insert("", "end", values=row)
tree.bind("<ButtonRelease-1>", on_treeview_select)
```



Рисунок 3.8 - Додані кнопки при виборі записів таблиці

При натисканні на frame лівою кнопкою миші зникають додаткові кнопки редагування, видалення. Виконується код:

```
def on_form(event):
    if not entry_find.get():
        entry_find.insert(0, "Назва продукції")
        entry_find.config(fg="gray") # Изменение цвета текста обратно на серый
    self.del_items.place_forget()
    self.change_items.place_forget()
    selected_items = tree.selection()
    if selected_items:
        tree.selection_remove(tree.selection())
self.zapas_menu.bind("<ButtonRelease-1>", on_form)
```

Пошук виконується за допомогою поля entry\_find і кнопки btn\_find, яка виконує функцію find(), код якої наведений в додатку Е

Склад №1 Запаси товарів

Пошук продукту: халва

№	Назва продукції	Кількість палетів	Ціна за од	Зона зберігання	Склад	Артикул	Категорія
810	халва	2	3	їжа	Склад №1	4TRHL	алкоголка

Рисунок 3.9 - Результат роботи пошуку

Таблиця має сортувати дані. Код функції сортування наведений в додатку Е. Функція перевіряє, чи стовпець є числом. Якщо так, то дані сортуються як числа. В іншому випадку дані сортуються як рядки.

Далі дані сортуються всі рядки у відповідному порядку (за зростанням або спаданням), а потім переміщуються у віджеті Treeview у відповідний порядок.

Функція викликається за натиском назви стовпця.

Склад №1 Запаси товарів

Назва продукції

№	Назва продукції	Кількість палетів	Ціна за од	Зона зберігання	Склад	Артикул	Категорія
810	халва	2	3	їжа	Склад №1	4TRHL	алкоголка
876	efef2	0	3	зона б1	Склад №1	1PM7F	алкоголка
844	уауа34	9	3	зона б1	Склад №1	1WJ8Z	алкоголка
737	уауа	241	3	зона б1	Склад №1	84433	папалам
776	каван	2	3	зона б1	Склад №1	YUJ0X	алкоголка
773	Барбарис	0	5	їжа	Склад №1	R1NIC	категорія А
772	Абрикос	51	10	їжа	Склад №1	CKOQC	фрукт
774	Лимон	2	22	холоднаа	Склад №1	J1WTO	категорія А
770	сало	14	22	їжа	Склад №1	C6LKC	папалам
775	масло	54	23	холоднаа	Склад №1	L0AD4	папалам
771	Хліб	1	32	нова зона	Склад №1	41097	незнаю

Рисунок 3.10 - Результат виконання сортування за колонкою «Ціна за од»

Також є функція очищення всієї таблиці:

```
def delete_all_data():
```

```
    try:
```

```
        execute_sql_query_insert(conn_str, "DELETE FROM Продукція")
```

```
        self.open_zapas_menu()
```

```
    except:
```

```
        delete_error(self.zapas_menu)
```

```
start_1 = Button(self.zapas_menu, image=self.delete_all, command=delete_all_data)
```

```

start_1.place(x=25, y=63)
start_1.config(background="White", highlightbackground="White",
activebackground="white")
ToolTip(start_1, text="Очистити таблицю")

```

ToolTip – створений клас для відображення підказки при наведенні курсором на графічний об’єкт. Результат роботи класу наведений на рисунку 3.11

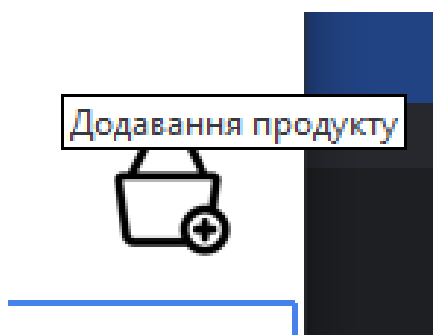


Рисунок 3.11 - Результат виконання роботи класу ToolTip

Автоматизовано систему друку даних відповідного фрейму. На фреймі розташована кнопка друку, яка викликає функцію виводу всіх даних в pdf-файл  
Код кнопки друку:

```

print_doc = Button(self.zapas_menu, image=self.printer, command=print_to_pdf)
print_doc.place(x=87, y=63)
print_doc.config(background="White", highlightbackground="White",
activebackground="white")
ToolTip(start_1, text="Друк")

```

Код функції print\_to\_pdf, яка підготовлює дані до друку наведено в додатку Е.

Автотранспорт			
№	Назва	Номер	Вантажопідйомність
105	BMW	FF3434DF	344
106	уауа	2323	3434

Рисунок 3.12 - Результат виконання функції print\_to\_pdf

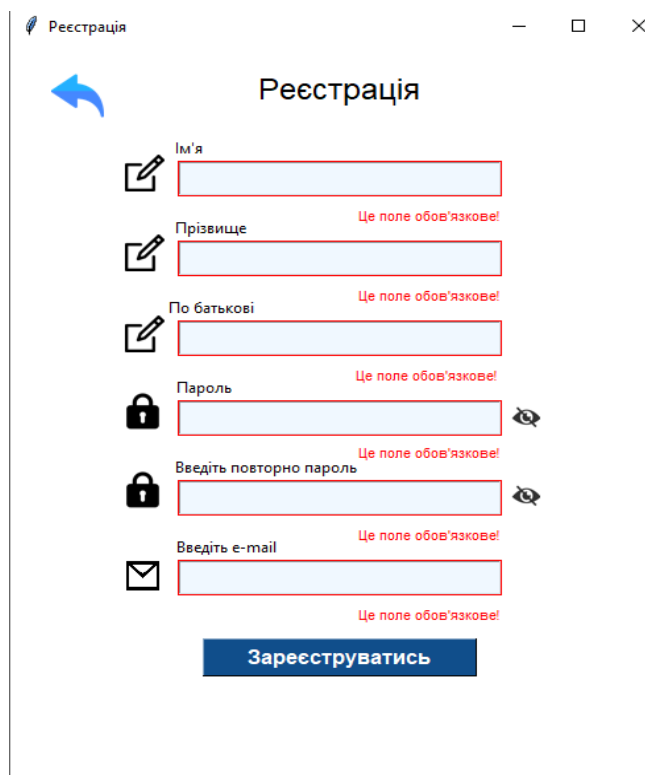
За таким принципом працюють усі інші фрейми (фрейм надходження, замовника, ТТН, водія тощо).

### 3.3. Інструкція користувача

Після запуску програми з'являється вікно авторизації, де користувач повинен увійти у свій обліковий запис за допомогою введення даних у поля (Рисунок 3.2 - ).

Якщо користувач не зареєстрований, то він може це зробити, натиснувши на «реєстрація» і відкриється вікно реєстрації (Рисунок 3.3 -).

Для успішної авторизації чи реєстрації потрібно заповнити всі поля введення даних, інакше виникне помилка про пусті поля:



The screenshot shows a registration window titled "Реєстрація". It contains several input fields, each with a red border and a red error message below it: "Це поле обов'язкове!". The fields are: "Ім'я", "Прізвище", "По батькові", "Пароль", "Введіть повторно пароль", and "Введіть e-mail". Each field has an icon to its left: a pencil for the first three, a lock for the passwords, and an envelope for the email. A blue button labeled "Зареєструватись" is at the bottom. The window title bar shows "Реєстрація" and standard window controls.

Рисунок 3.13 - Обробка помилки незаповнених полів

Якщо при реєстрації відбудеться введення двох різних паролів, то виникне така помилка:

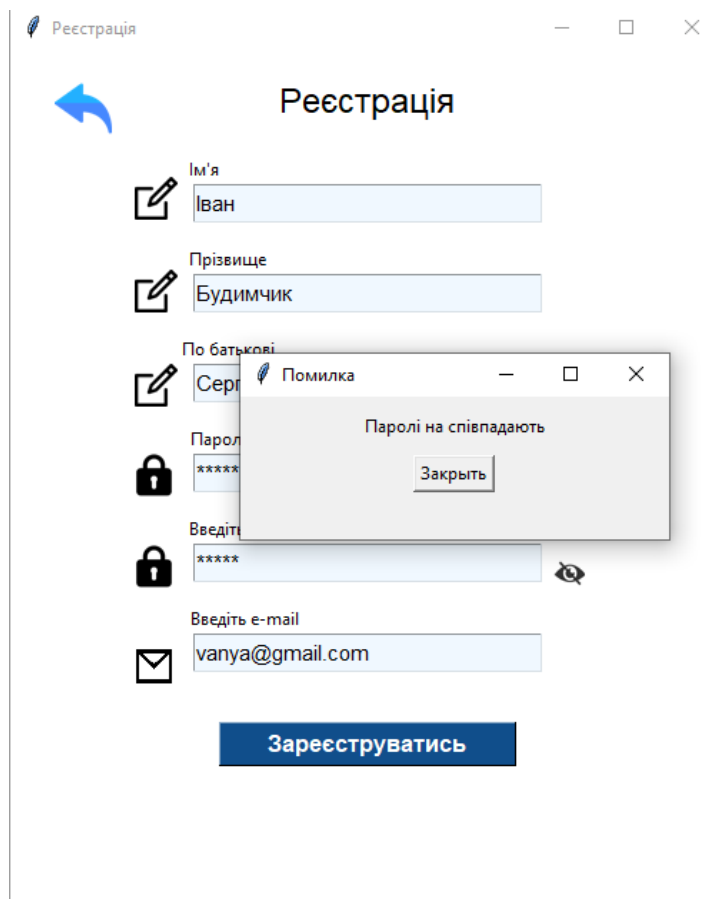


Рисунок 3.14 - Обробка помилки при введенні двох різних паролів  
Якщо ж реєстрація пройшла успішно виводиться вікно авторизації:

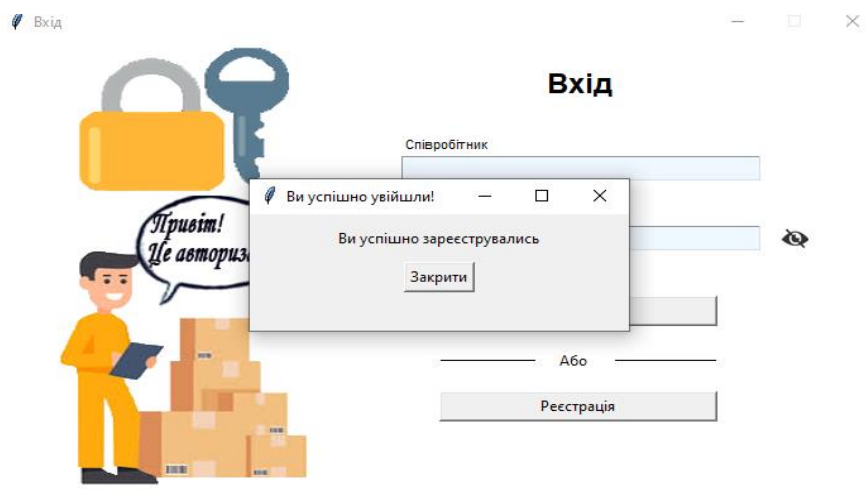


Рисунок 3.15 - Успішна реєстрація

Вікно авторизації і реєстрації має можливість приховувати і показувати пароль, що є корисним для забезпечення безпеки та зручності.

Співробітник  
Іван Будимчик Сергійович

Пароль  
\*\*\*\*\*

Увійти

Або

Реєстрація

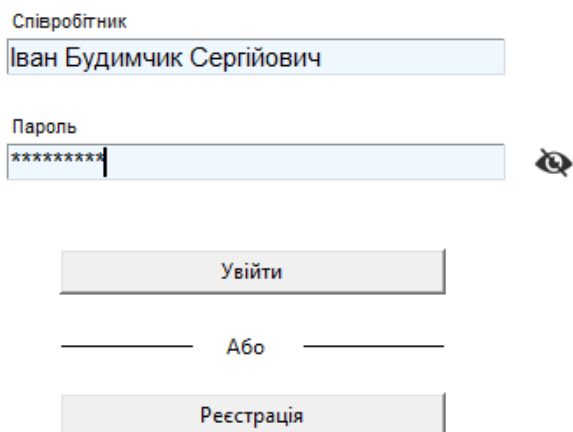
The image shows a login interface. At the top, there is a label 'Співробітник' (Employee) above a text input field containing the name 'Іван Будимчик Сергійович'. Below this is a label 'Пароль' (Password) above a text input field filled with asterisks '\*\*\*\*\*'. To the right of the password field is an eye icon with a slash through it, indicating that the password is hidden. Below the password field are three buttons: 'Увійти' (Login), 'Або' (Or), and 'Реєстрація' (Registration).

Рисунок 3.16 - Прихований пароль

## Вхід

Співробітник  
Іван Будимчик Сергійович

Пароль  
111\_ваня\_111

Увійти

Або

Реєстрація

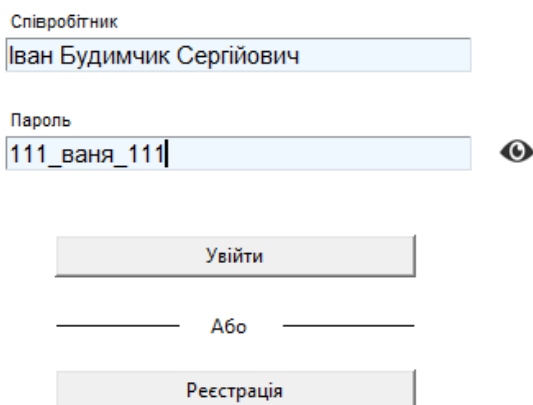
The image shows the same login interface as in Figure 3.16, but with the password field visible. The password '111\_ваня\_111' is entered in the text input field. The eye icon is now an open eye, indicating that the password is visible. The other elements, including the employee name and the 'Увійти', 'Або', and 'Реєстрація' buttons, remain the same.

Рисунок 3.17 - Відкритий пароль

Після успішної авторизації, при відсутності інформації про наявність складів, виводиться повідомлення з пропозицією додати новий склад підприємства.

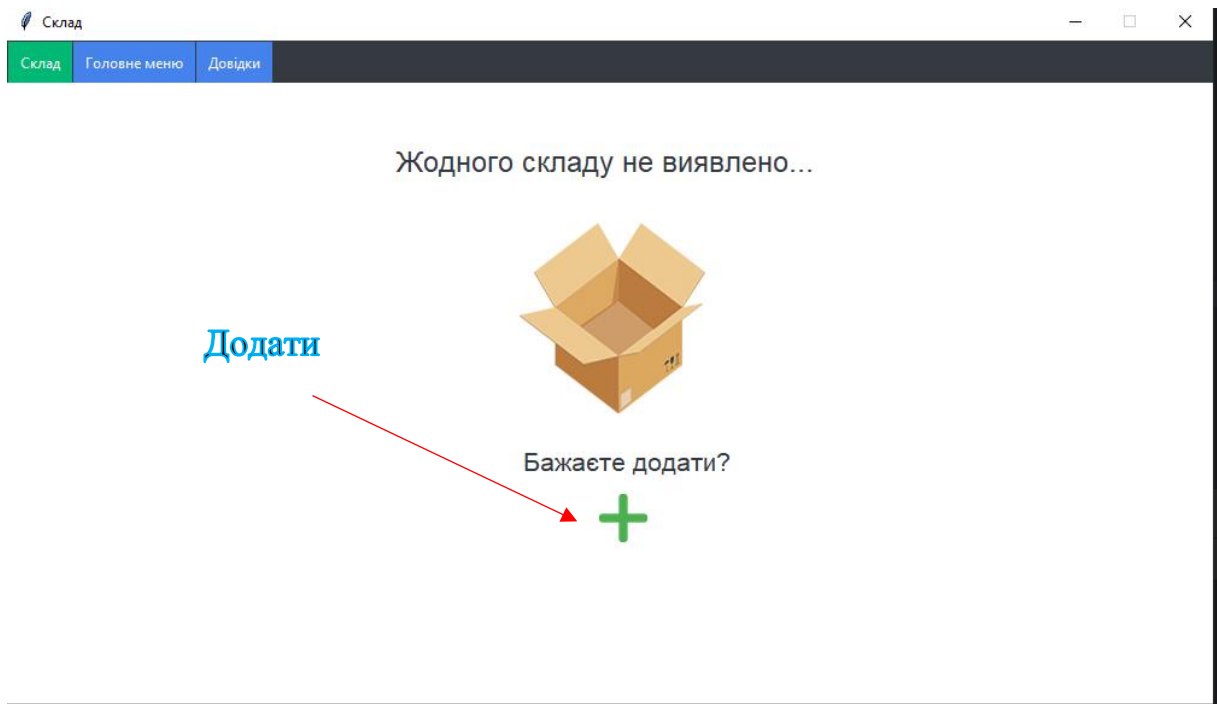


Рисунок 3.18 - Додавання нового складу підприємства

Натиснувши на «Плюс» або «Додати», відкриється вікно додавання складу:

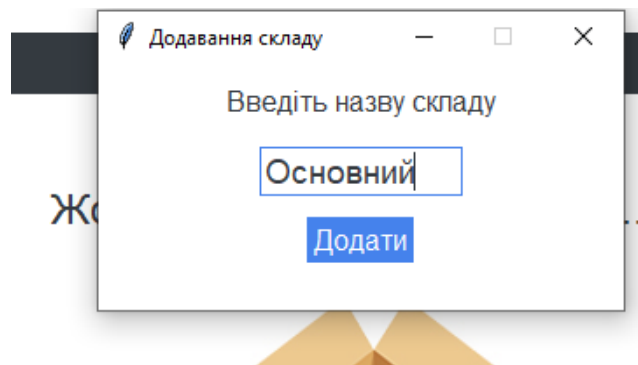


Рисунок 3.19 - Вікно додавання складу

Введіть склад і натисніть «Додати». Після цього відкриється вікно вибору наявних складів з можливістю обирати склад і додавати:

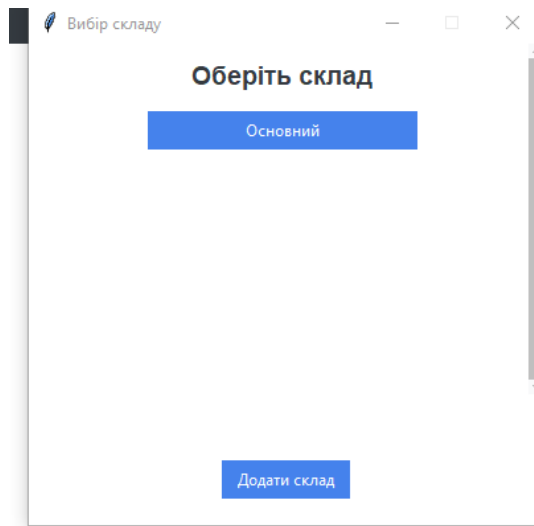


Рисунок 3.20 - Вікно вибору складу

Після натискання на обраний склад відкриється головне меню інформаційної системи (Рисунок 3.4 -)

Головне меню містить такі графічні об'єкти або функції:

- верхнє меню, яке містить три кнопки: «Склад» - викликає каскадне меню для вибору або додавання складу (**Error! Reference source not found.**). Кнопка «Головне меню», натиснувши на яку користувач перейде до головного меню з будь-якого відкритого пункту. Третя кнопка «Довідник», яке викликає теж каскадне меню, які містить додаткові функції перегляду авто, перевізників, категорії продукції і їх зони зберігання (Рисунок 3.6 -);
- кнопку видалення поточного складу (Рисунок 3.22 -);
- кнопку редагування назви поточного складу (Рисунок 3.22 -);
- перегляд запасів – кнопка «Запаси товарів»;
- додавання надходження товарів – кнопка «Надходження»;
- продаж товарів – кнопка «Продаж»;
- списання товарів – кнопка «Списання»;
- перегляд клієнтів, замовників – кнопка «Клієнти»;
- формування ТТН – кнопка «ТТН».



### Рисунок 3.22 - Кнопки видалення і редагування складу

Натиснувши «Запаси товарів» відкривається форма із даними про продукцію:

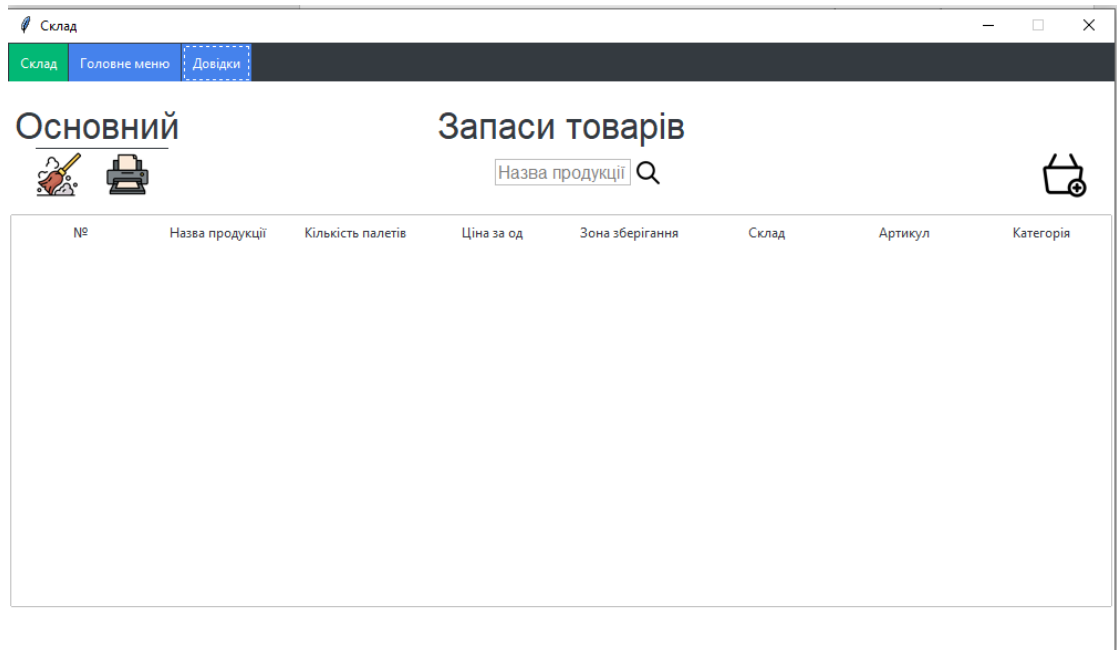


Рисунок 3.23 - Вікно «Запаси товарів»

Форма містить таблицю, яка відображає дані про продукцію. Кнопку додавання нової продукції, пошуку продукції, очищення всіх даних і друку всіх товарів.

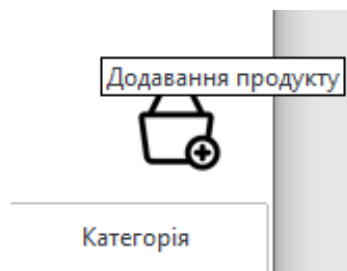


Рисунок 3.24 - Кнопка додавання нового товару

Натиснувши на кнопку додавання, відкривається вікно додавання продукції:

Редагування продукту

**Додавання продукту**

Назва продукту:

Кількість(палетів):

Категорія:

Ціна (за од.):

Зона зберігання:

**Додати**

Рисунок 3.25 - Вікно додавання нового товару

У вікні ви можете додати нову категорію для продукції або зону зберігання, натиснувши на «плюс»:

Нова категорія

**Додавання категорії**

Назва категорії:

**Додати**

Рисунок 3.26 - Вікно додавання нової категорії

Зона зберігання

**Нова зона зберігання**

Назва зони зберігання:

Місткість палетів:

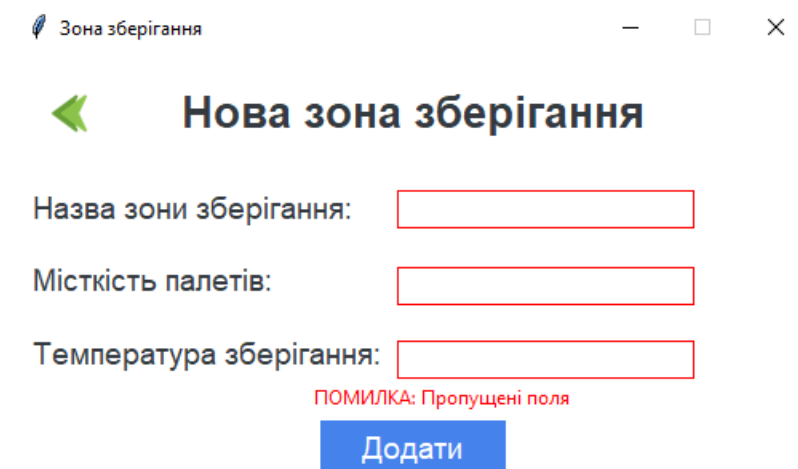
Температура зберігання:

**Додати**

Рисунок 3.27 - Вікно додавання нової зони зберігання

Також категорію, зону зберігання можна додати у верхньому меню, натиснувши на кнопку «Довідки»

Усі форми мають спільний характер. Обробка помилок введення такі:



Зона зберігання

### ← Нова зона зберігання

Назва зони зберігання:

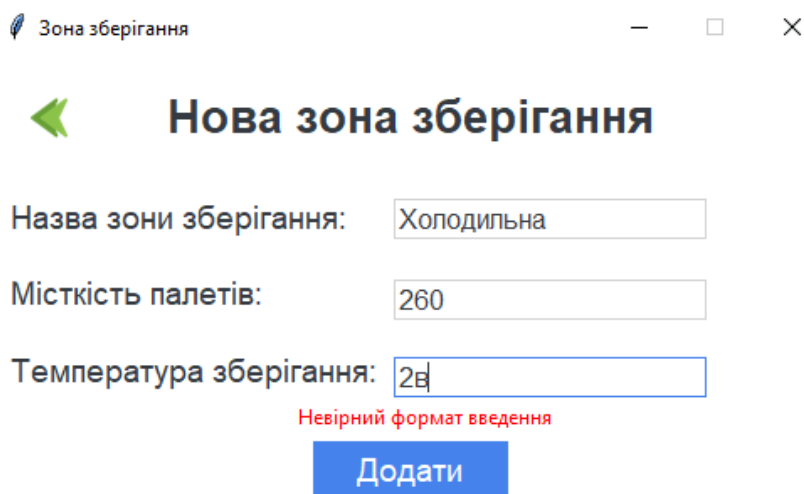
Місткість палетів:

Температура зберігання:

ПОМИЛКА: Пропущені поля

Додати

Рисунок 3.28 - Помилка незаповнених полів



Зона зберігання

### ← Нова зона зберігання

Назва зони зберігання:

Місткість палетів:

Температура зберігання:

Невірний формат введення

Додати

Рисунок 3.29 - Помилка некоректного введення формату даних

Після коректного введення даних вони додаються до таблиці. Приклад доданого товару до зон зберігання:

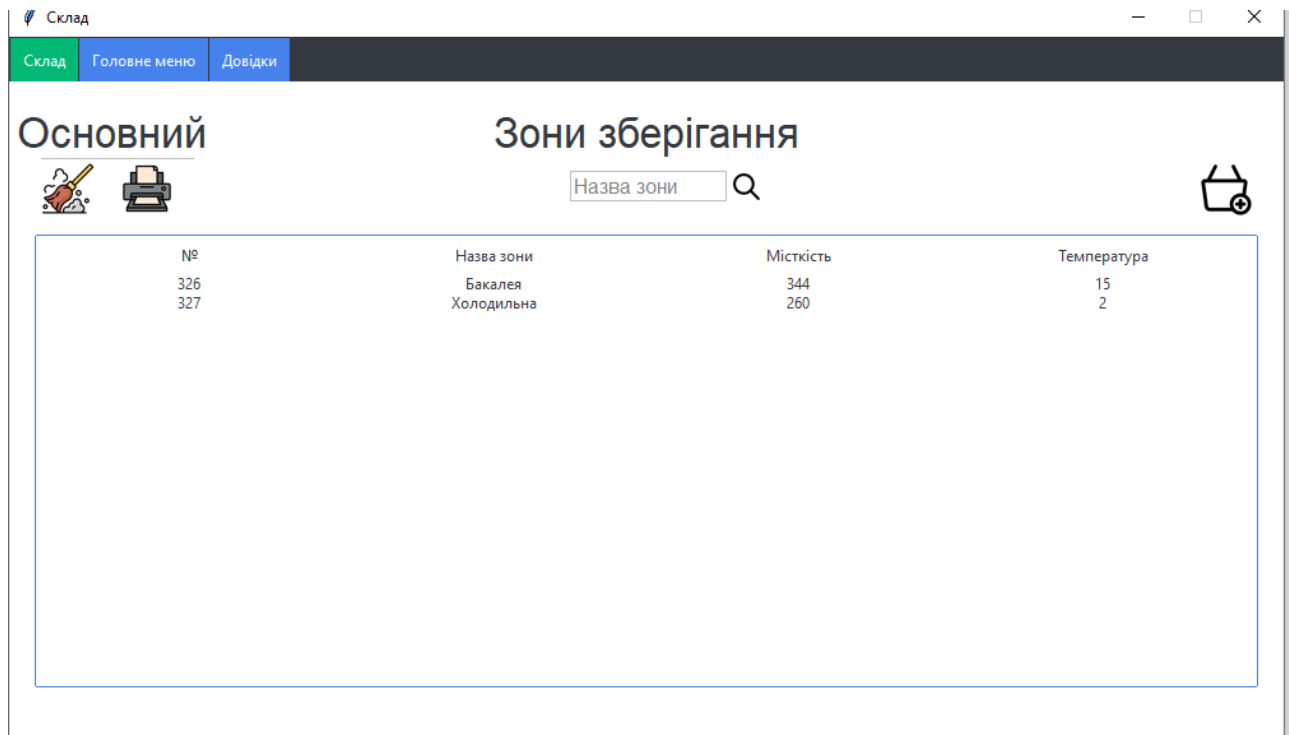


Рисунок 3.30 - Додані зони зберігання

Такий функціонал мають всі меню інформаційної системи.

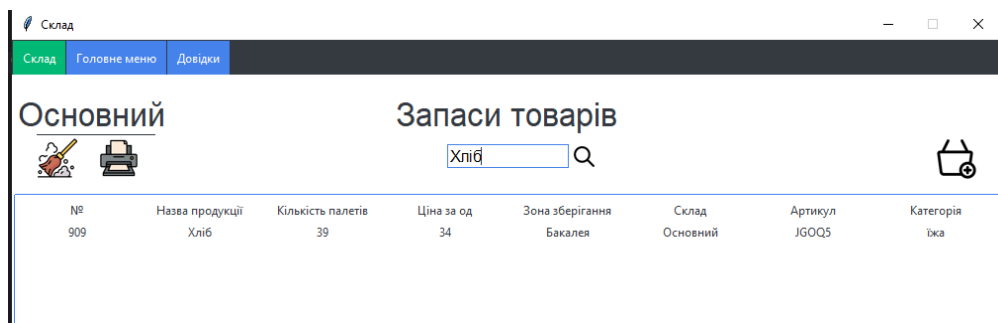


Рисунок 3.31 - Шукані дані в таблиці «продукція»


Для пошуку товару необхідно ввести назву товару і натиснути на рисунок лупи.

Перейдемо до функції «Надходження». Розглянемо приклад додавання нового надходження товару:


Нове надходження

### Додавання надходження

Номер ордеру:


Назва продукту:  

Кількість:

Зона зберігання:  

Цех:

Номер партії:

Дата:  

Сума:

Рисунок 3.32 - Додавання надходження


Обираємо товар, який надходить, заповнюємо дані і натискаємо «Додати». Дані надходження зберігаються в таблицю «Надходження» (Рисунок Д.3 -). Доданий товар оновився, а саме його кількість:


Склад

Склад | Головне меню | Довідки

## Основний

## Запаси товарів

Назва продукції 



№	Назва продукції	Кількість палетів	Ціна за од	Зона зберігання	Склад	Артикул	Категорія
909	Хліб	39	34	Бакалея	Основний	JG0Q5	їжа

Рисунок 3.33 - Оновлений товар

Так само працює продаж і списання товару, тільки кількість товару відповідно зменшується.

Для редагування і видалення необхідно обрати відповідний рядок. Редагувати і видаляти запис можна, обравши запис. Натиснувши на рядок в таблиці, додадуться дві кнопки:

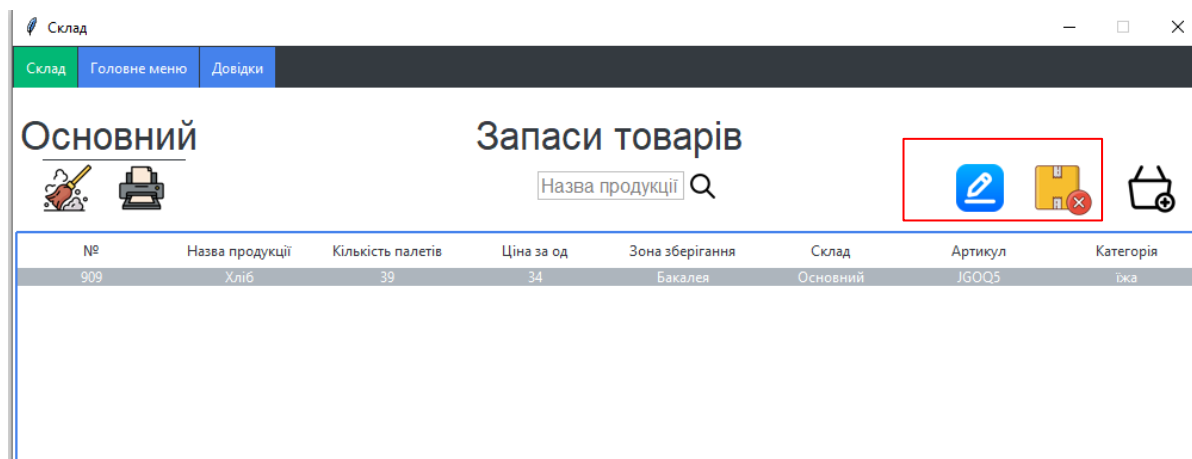


Рисунок 3.34 - Додані кнопки редагування і видалення

Редагування відкриє вікно із даними, а кнопка видалення видаляє рядок

Редагування продукту

**Редагування продукту**

Назва продукту:

Кількість(палетів):

Категорія:  (+)

Ціна (за од.):

Зона зберігання:  (+)

Рисунок 3.35 - Форма редагування запису з меню «Запаси»

Стиль і принцип редагування даних на всіх формах подібний рисунку вище.

Зліва розташовані дві кнопки: кнопка очистки всіх даних і кнопка друку відповідно.

Після очистки всіх даних вся таблиця очиститься. Роботу функції друку продемонструємо у вікні створення ТТН.

Додавши замовника, водія і транспорт можна формувати ТТН

Із форми додавання ТТН можна додавати й інші дані.

Рисунок 3.36 - Формування ТТН

Після додавання ТТН можна її роздрукувати. Є два види друку:

1. Друк всіх ТТН як загальний документ
2. Друк обраної ТТН



Рисунок 3.37 - Кнопка друку всіх ТТН

Результат друку:

#### Товаро-транспортна накладна

№	Номер	Склад	Замовник	Продукція	Кіл-ть	Ціна/од	Сума	Водій	Транспорт	Дата
146	1	Основний	Вася. П.В.	Хліб	3	34	102	Булій Д.Ю.	BMW	2024-05-15

Рисунок 3.38 - Результат друку всіх ТТН

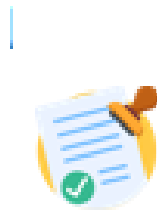


Рисунок 3.39 - Кнопка друку обраної ТТН

Результат друку:

Товаро-транспортна накладна

Дата: 2024-05-15

Номер ТТН 1

Замовник: Вася. П.В.

Відправник: ПрАТ 'Оболонь'

Адреса: уауауа

Автомобіль: BMW

Водій: Булій Д.Ю.

Відомості про товар

№	Продукція	Кіл-ть	Ціна/од	Сума
146	Хліб	3	34	102

Номер водія: 34343

Рисунок 3.40 - Результат друку обраної ТТН

## РОЗДІЛ 4. ОХОРОНА ПРАЦІ

### 4.1. Охорона праці на підприємстві

ПрАТ «Оболонь» підтвердила відповідність вимогам ДСТУ ISO ДСТУ-П OHSAS 18 001:2006 (Системи управління безпекою та гігієною праці) [1].

Охорона праці на підприємстві включає в себе ряд заходів, спрямованих на забезпечення безпеки та здоров'я працівників:

- розробку та впровадження програми охорони праці, яка визначає процедури та вимоги щодо безпеки та здоров'я на робочому місці;
- проведення регулярних інструктажів з безпеки та ознайомлення працівників з правилами та процедурами безпеки;
- забезпечення необхідного оснащення та засобів індивідуального захисту (зокрема, спеціальних одягу, взуття, респіраторів тощо);
- проведення регулярних перевірок та обслуговування обладнання і машин з метою попередження аварій і травматизму;
- впровадження процедур дії в екстрених ситуаціях, наприклад, у випадку пожежі або аварії, включаючи план евакуації та навчання персоналу щодо його виконання;
- моніторинг робочих умов та їх впливу на здоров'я працівників, з метою вчасного виявлення та усунення потенційних ризиків;
- систематичний аналіз нещасних випадків та інцидентів на робочому місці з метою вжиття заходів для їх попередження у майбутньому;
- залучення працівників до процесу управління безпекою та залучення їх пропозицій та зауважень для поліпшення системи охорони праці.

На підприємстві впроваджено протипожежну безпеку, яка є критично важливою для забезпечення безпеки працівників, майна і довкілля. Основні аспекти протипожежної безпеки включають:

1. розробку та впровадження плану пожежної безпеки: ідентифікацію потенційних загроз пожежі, розробку процедур евакуації, визначення ролей та відповідальності персоналу у разі пожежі та проведення навчання з протипожежної

безпеки;

2. установку та обслуговування протипожежного обладнання: встановлення пожежних тривожних систем, пожежних гідрантів, вогнегасників та іншого протипожежного обладнання. Це обладнання повинно періодично перевірятися та обслуговуватися, щоб забезпечити його ефективність;

3. організацію пожежних вправ: регулярні пожежні вправи допомагають підготувати персонал до дій у разі пожежі. Ці вправи повинні включати в себе практичні симуляції евакуації, використання протипожежного обладнання та координацію дій персоналу;

4. запобігання пожежам: регулярну перевірку електричних систем, використання негорючих матеріалів у будівлях та обмеження використання відкритого вогню на робочих місцях;

5. співпрацю з місцевими пожежними відділами: підприємство повинно мати контакт із місцевими пожежними службами та встановлені процедури співпраці у разі виникнення пожежі або іншої надзвичайної ситуації.

## ВИСНОВОК

В даній кваліфікаційній роботі було розроблено інформаційну систему обліку готової продукції на складі ПрАТ «Оболонь». Для розробки інформаційної системи було використано середовище PyCharm з використанням графічної бібліотеки tkinter та СУБД PostgreSQL. У результаті розробки ІС було створено програмний комплекс, який призначений для автоматизації та оптимізації управління складським господарством підприємства. Основна мета розробки ІС полягає у поліпшенні ефективності управління запасами та оптимізації всіх процесів, пов'язаних із складською діяльністю.

В процесі проектування і розробки інформаційної системи було успішно реалізовано всі задачі, визначені на етапі планування, та реалізовано інформаційну систему, яка відповідає всім вимогам та критеріям, визначеним у технічному завданні. Розроблена інформаційна система може успішно вирішувати ряд таких задач як: контроль запасів продукції, управління запасами (система автоматизує процеси замовлення, отримання, зберігання, відстеження та відвантаження товарів), генерація звітів. Розроблена ІС може бути інтегрована з іншими системами управління підприємства.

Проаналізувавши способи використання інформаційної системи, було визначено можливий ефект від її впровадження. Річний прибуток від впровадження системи буде досягнуто за рахунок покращення ефективності роботи відділу, зменшенню кількості можливих помилок та витрат на їх виправлення і, орієнтовно, складатиме 10 000 грн на рік, а окупність ІС після її впровадження становитиме 8.1 років.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ОБОЛОНЬ [Електронний ресурс] – Режим доступу до ресурсу: <https://obolon.ua/ua>
2. CA ERwin Process Modeler PostgreSQL [Електронний ресурс] – Режим доступу до ресурсу: <https://www.broadcom.com/products/enterprise-software/erwin-data-modeler/process-modeler>
3. Вибір моделювання бізнес-процесів PostgreSQL [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.crmoz.com/blogs/post/business-process-modeling>
4. Нотація BPMN PostgreSQL [Електронний ресурс] – Режим доступу до ресурсу: <https://iampm.club/ua/blog/shho-take-bpmn-diagrama-i-navishho-vona-potribna-z-prikladami-2/>
5. Побудова бізнес-процесів за допомогою Bizagi PostgreSQL [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ba.in.ua/2022/11/25/typovi-pomylyky-pry-modelyuvanni-biznes-proczesiv-v-notacziyi-bpmn-ch-1/>
6. База даних PostgreSQL [Електронний ресурс] – Режим доступу до ресурсу: <https://www.guru99.com/uk/introduction-postgresql.html>
7. SQL-запити [Електронний ресурс] – Режим доступу до ресурсу: <https://foxminded.ua/sql-zapyty/>
8. Графічна бібліотека tkinter [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.python.org/uk/3/library/tkinter.html>
9. Python ООП [Електронний ресурс] – Режим доступу до ресурсу: <https://acode.com.ua/object-oriented-programming-python/>
10. Підручник з Python [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.python.org/uk/3/tutorial/index.html>
11. Дизайн графічної бібліотеки ttkbootstrap [Електронний ресурс] – Режим доступу до ресурсу: <https://ttkbootstrap.readthedocs.io/en/latest/>
12. М'якшило О.М. CASE-технології у проектуванні інформаційних систем: електронний навчальний посібник для студ. вищих навч. закладів / О.М. М'якшило, Л.Г. Загоровська, – К.: НУХТ, 2017. – 190 с.

13. Основні функції складу ООП [Електронний ресурс] – Режим доступу до ресурсу: <https://bitfaktura.com.ua/skladaska-prohrama/vsya-informatsiya>

14. Організація робіт зі створення інформаційних систем обліку. Стадії та етапи робіт зі створення та впровадження ІСО ООП [Електронний ресурс] – Режим доступу до ресурсу: <https://buklib.net/books/24427/>

15. Методичні рекомендації до виконання випускної кваліфікаційної роботи на здобуття освітнього ступеня «Бакалавр» спеціальності 122 «Комп'ютерні науки» освітньо-професійної програми «Комп'ютерні науки» ден. форми навчання [Електрон. ресурс] / уклад. О. М. М'якшило, М. П. Костіков. – К.: НУХТ, 2022. – 34 с.

16. Плєскач В. Л. Інформаційні системи і технології на підприємствах [Електронний ресурс] / В. Л. Плєскач, Т. Г. Затонацька // Знання. – 2011. – Режим доступу до ресурсу: [https://pidruchniki.com/1059110247701/informatika/informatsiyni\\_sistemi\\_i\\_tehnologiyi\\_na\\_pidpriyemstvah](https://pidruchniki.com/1059110247701/informatika/informatsiyni_sistemi_i_tehnologiyi_na_pidpriyemstvah).

17. ДСТУ 3918:1999 (ISO/IEC 12207:2008). Інформаційні технології. Процеси життєвого циклу програмного забезпечення. – 57 с.

18. ДСТУ 2941:1994. Системи оброблення інформації. Розроблення систем. Терміни та визначення.

19. ДСТУ 1.0:2003. СТД 1.0:2003. Національна стандартизація. Основні положення.

20. ДСТУ ISO/IEC 27000:2015. Інформаційні технології. Методи захисту. Система управління інформаційною безпекою. Огляд і словник.

21. ДСТУ ISO/IEC TR 15504. Інформаційні технології. Оцінювання процесів життєвого циклу програмних засобів. – 315 с.

22. Пасічник В. В., Резніченко В. А. Організація баз даних та знань. – К. : ВНУ, 2006. – 384 с.

23. Інструментальне середовище ERWin. Вивчення інтерфейсу [Електронний ресурс] // Національний університет Львівська політехніка. – 2020. – Режим доступу до ресурсу: [http://antibotan.com/file.html?work\\_id=516530](http://antibotan.com/file.html?work_id=516530).

24. Об'єктно-орієнтоване програмування мовою PYTHON : Конспект лекцій [Електронний ресурс] : навч. посіб. для студ. спеціальності 153 «Мікро- та наносистемна техніка» освітньої програми «Мікро- та наноелектроніка» / КПІ ім. Ігоря Сікорського ; уклад.: Д. Д. Татарчук, Ю. В. Діденко. – Електронні текстові данні (1 файл: 1,3 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 129 с.

25. Раєвська Н. Огляд безкоштовних і платних аналогів програми 1С:Підприємство, як альтернатива: є вона, чи все ж ні? [Електронний ресурс] / Наталя Раєвська // Блог. – 2018. – Режим доступу до ресурсу: <https://tqm.com.ua/ua/likbez/ua-articles/bezkoshtovni-analohy-1s#bukkyper>.

26. Гайна Г. А. Основи проектування баз даних : навч. посіб. – К. : КНУБА, 2005. – 204 с.

27. Конспект лекцій з дисципліни [Електронний ресурс] "Проектування інформаційних систем" для студентів спеціальності 122 "Комп'ютерні науки" /укл. М'якшило О.М., Харкянен О.В.- К.:НУХТ, 2018 – 48 с.

28. Управління ІТ проектами [Електронний ресурс]: методичні рекомендації до самостійної роботи для здобувачів освітнього ступеня «Бакалавр» спеціальності 122 «Комп'ютерні науки» освітньо- професійних програм «Комп'ютерні науки» та «Інформаційні системи та штучний інтелект» денної та заочної форм навч./уклад. С. В. Грибков.

29. Державна служба України з питань праці. Охорона праці на підприємстві – Режим доступу до ресурсу: <https://te.dsp.gov.ua/ohorona-pratsi-na-pidpryyemstvi-shho-potribno-znaty/>

30. ДСТУ 3008-2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання;

## ДОДАТКИ

## ДОДАТОК А. ОРГАНІЗАЦІЙНА СТРУКТУРА ПРАТ «ОБОЛОНЬ»

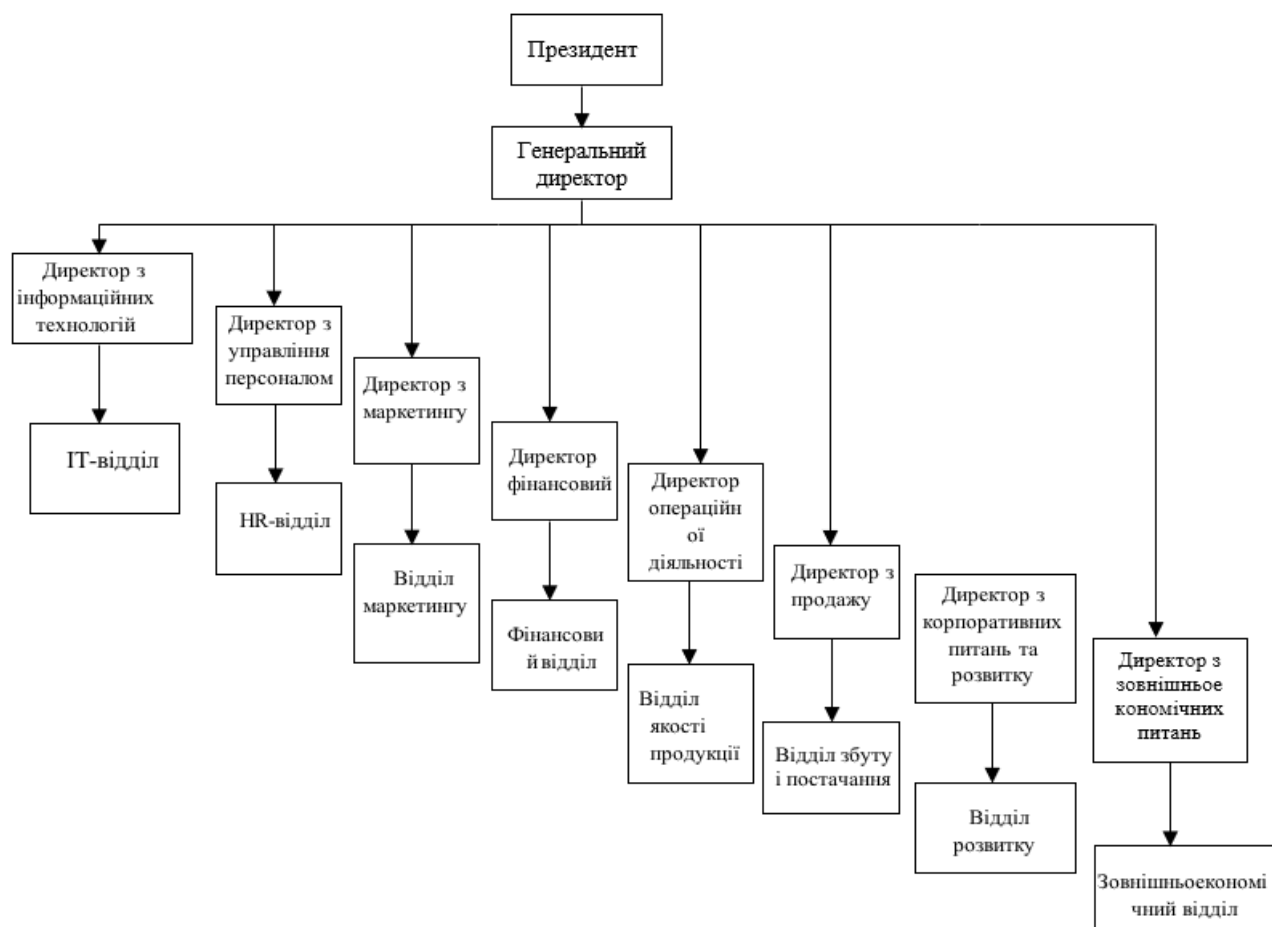


Рисунок А.1 – Організаційна структура ПрАТ «Оболонь»

## ДОДАТОК Б. ФУНКЦІОНАЛЬНІ МОДЕЛІ ДІЯЛЬНОСТІ ПРАТ «ОБОЛОНЬ»

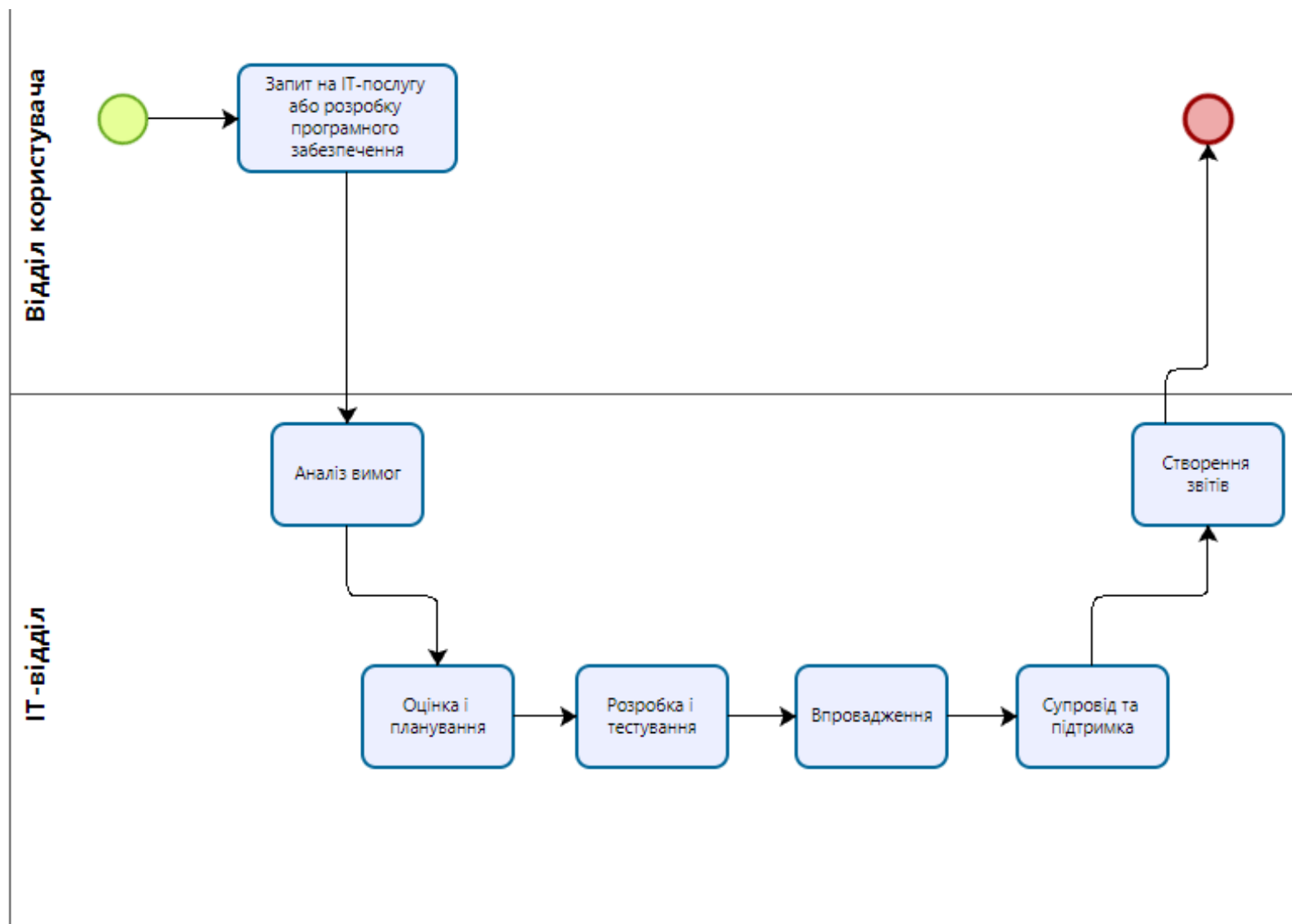


Рисунок Б.1 - Функціональна модель ІТ-відділу ПрАТ «Оболонь»

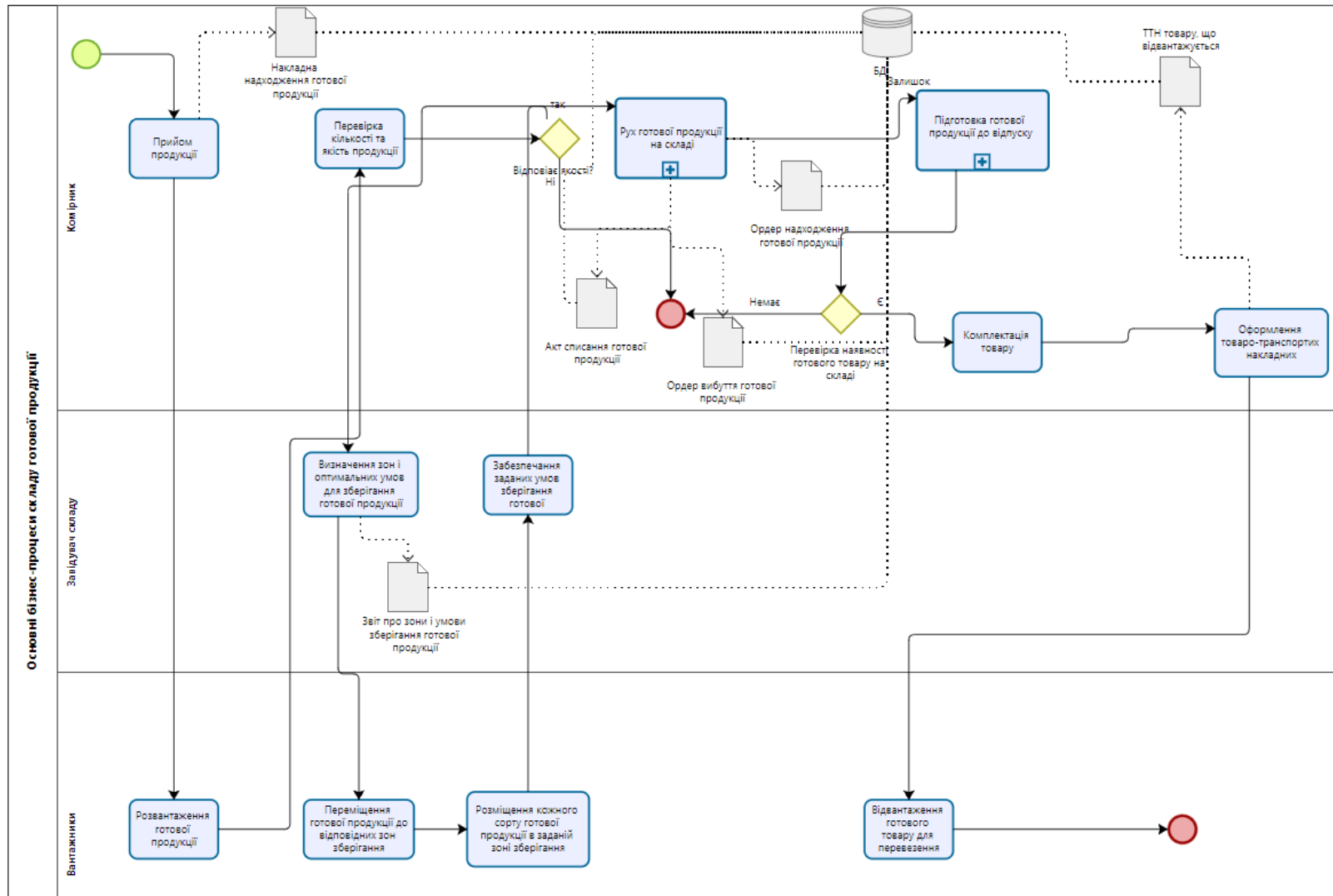


Рисунок Б.2 - Функціональна модель складу готової продукції

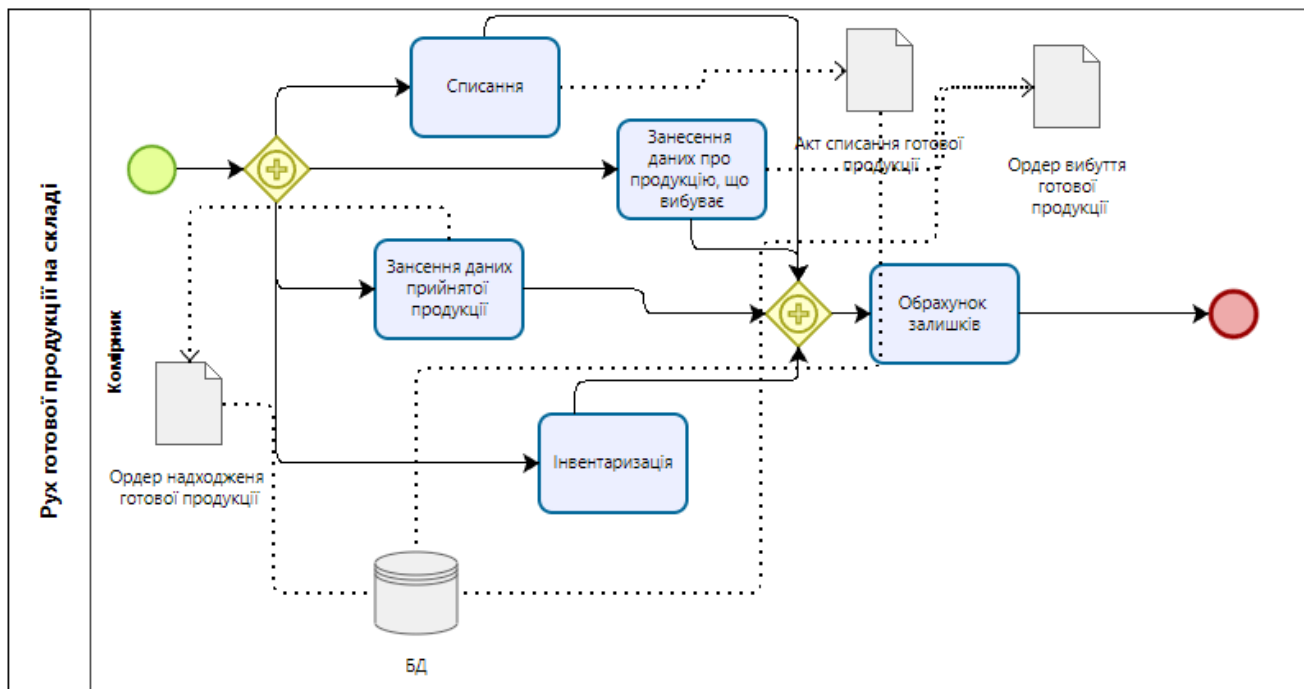


Рисунок Б.3 - Модель підпроцесу «процес прийому, обліку і зберігання готової продукції (1 рівень декомпозиції)»

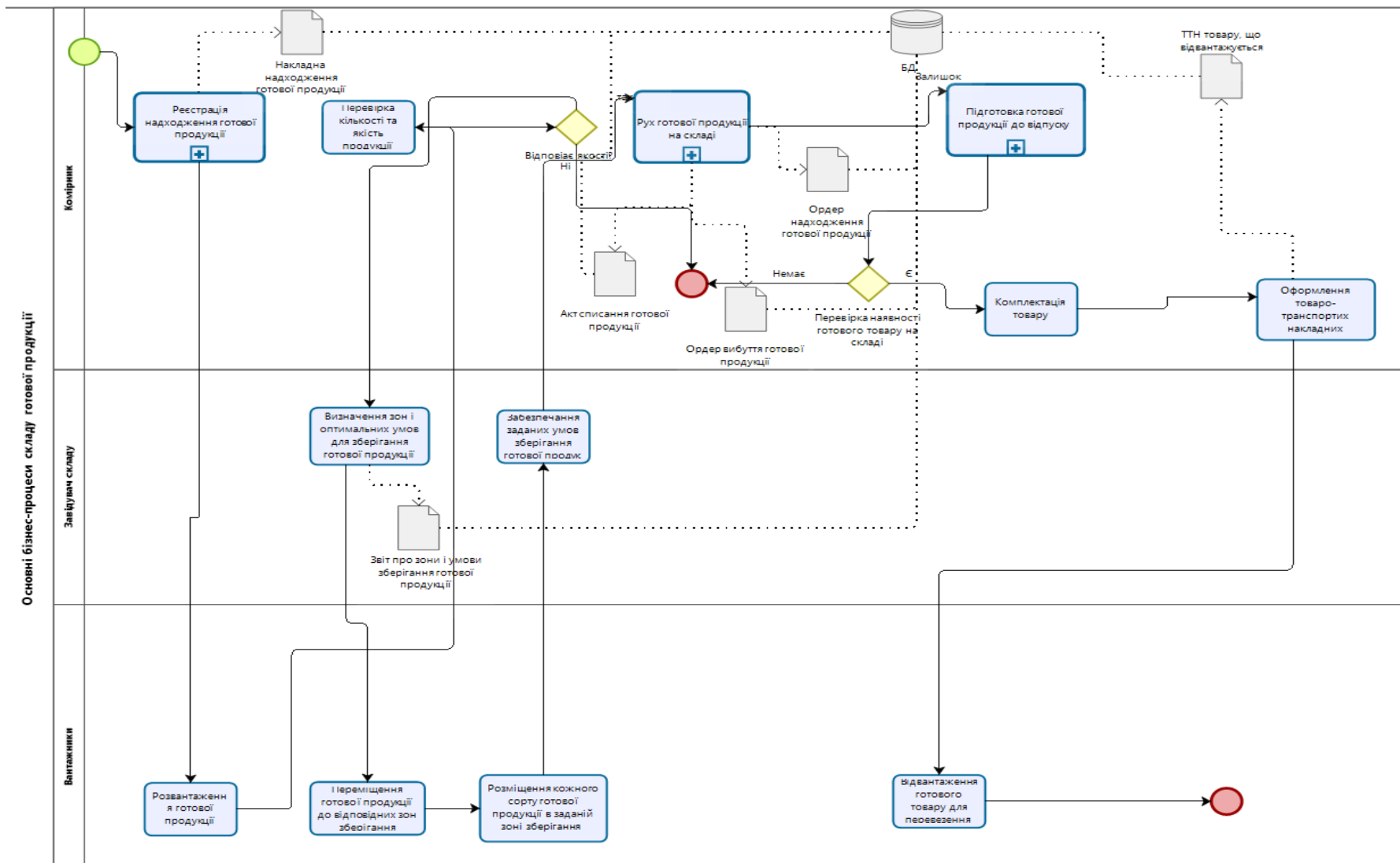


Рисунок Б.4 - Модель складу готової продукції ТО-ВЕ

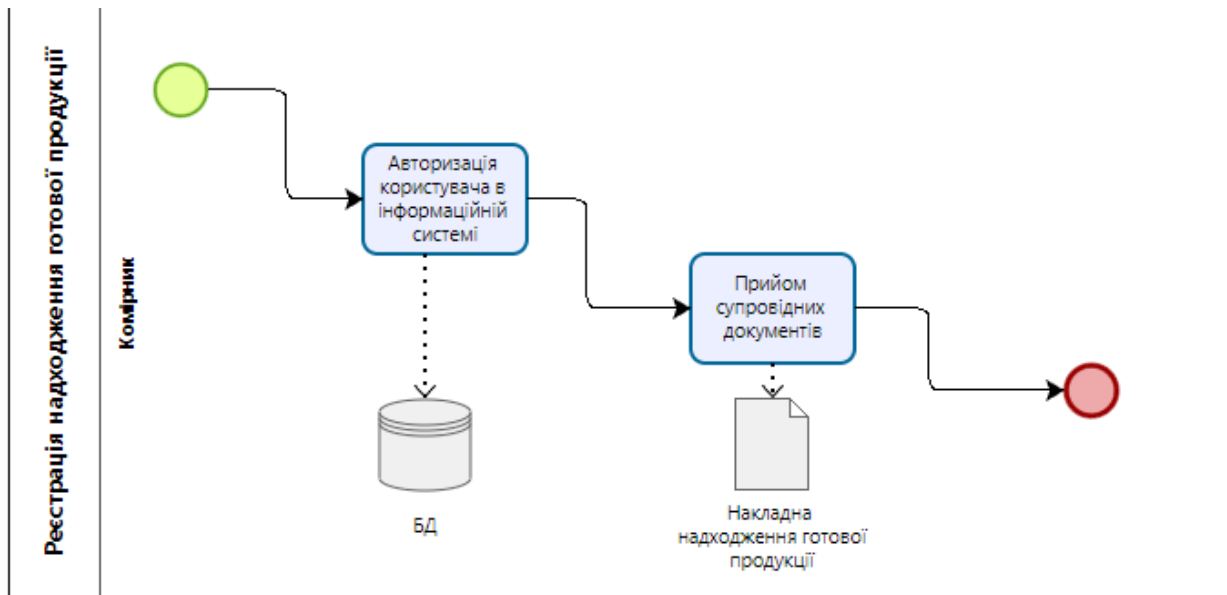


Рисунок Б.5 - Модель підпроцесу «реєстрація надходження готової продукції (1 рівень декомпозиції)»

## ДОДАТОК В. ЛОГІЧНА МОДЕЛЬ БД

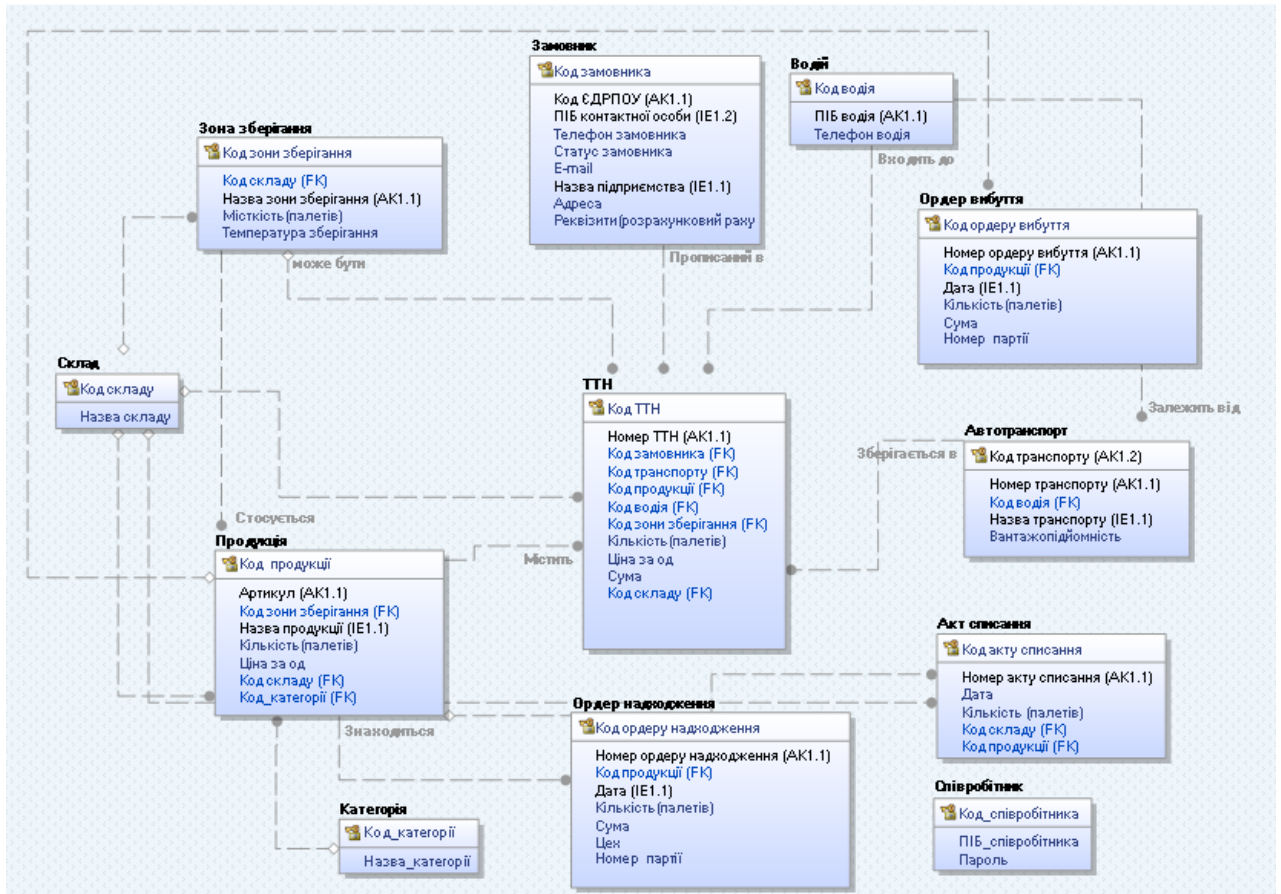


Рисунок В.1 - Логічна схема бази даних

## ДОДАТОК Ж. ФІЗИЧНА МОДЕЛЬ БД

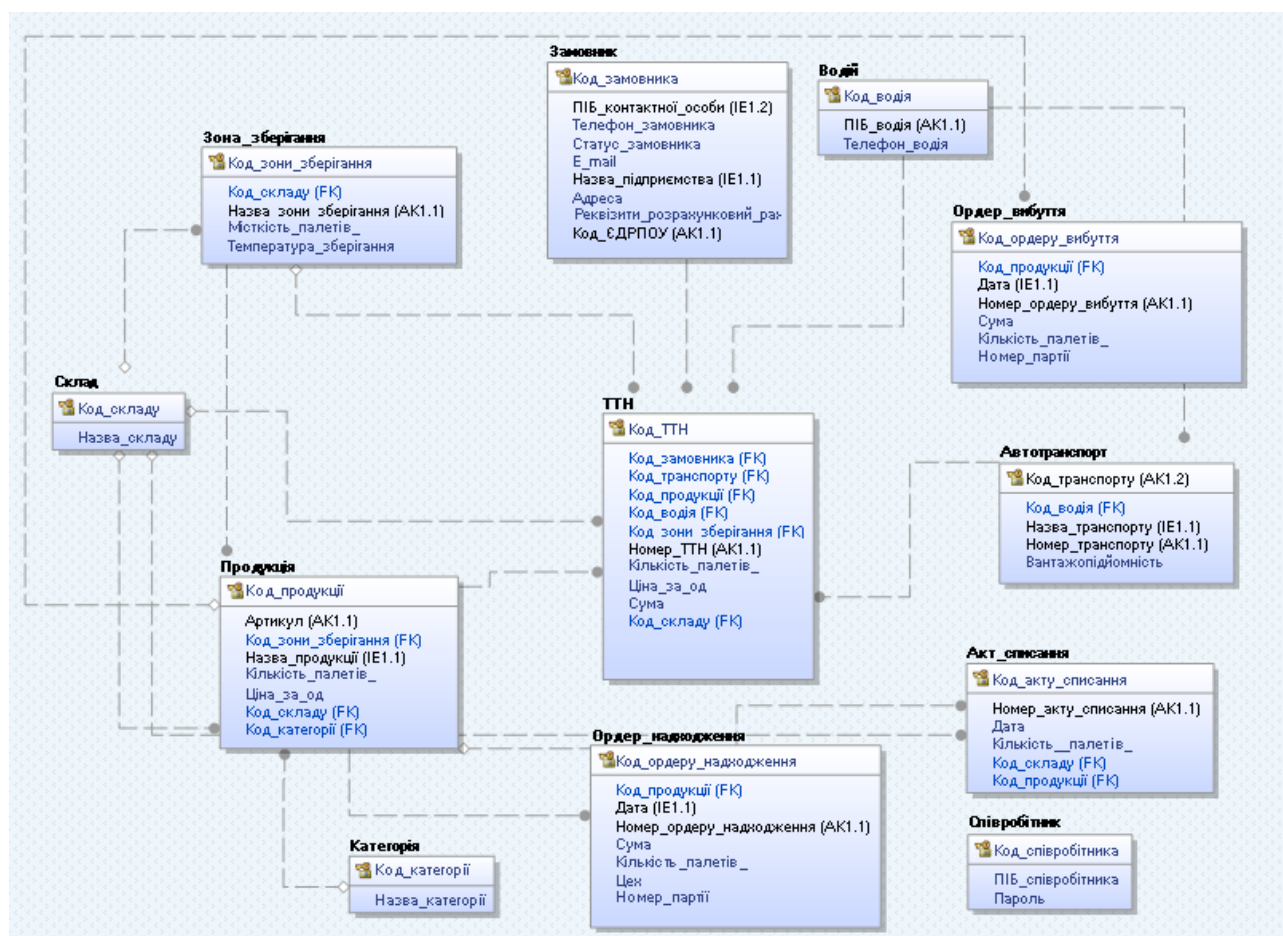


Рисунок Ж.1 - Фізична схема бази даних

## ДОДАТОК Д. ВІДЕОКАДРИ ІНТЕРФЕЙСУ

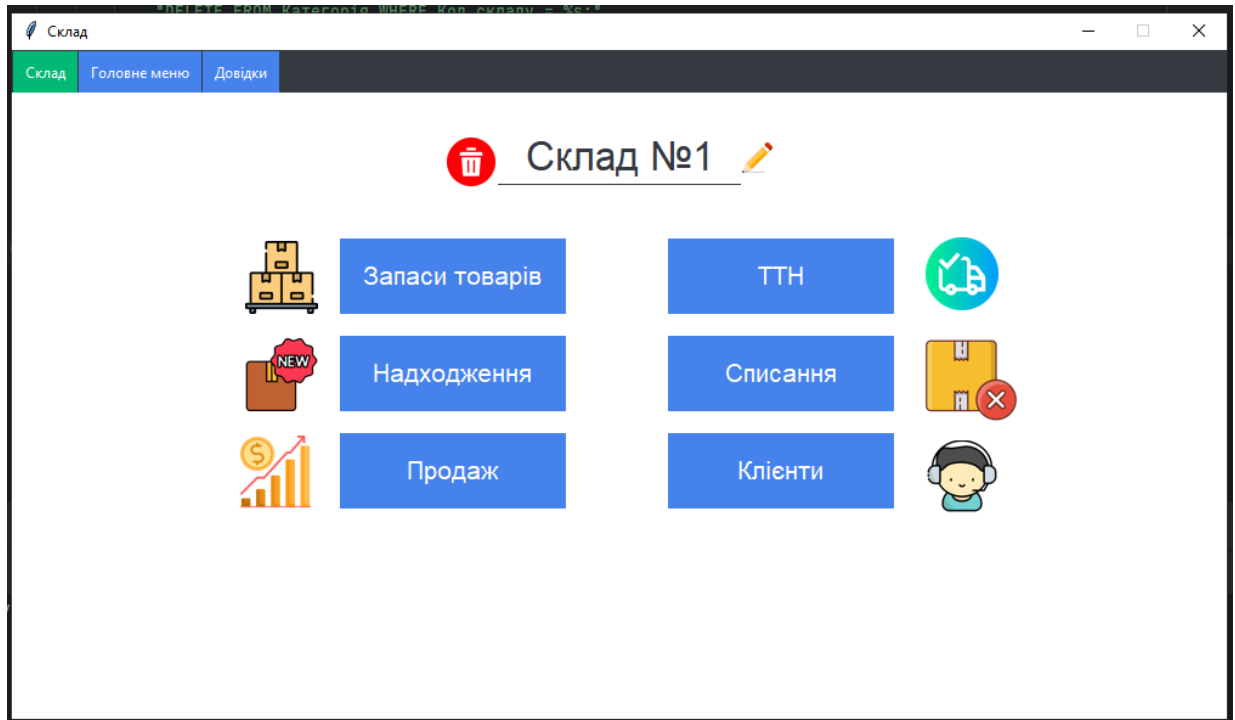


Рисунок Д.1 - Головне меню інформаційної системи

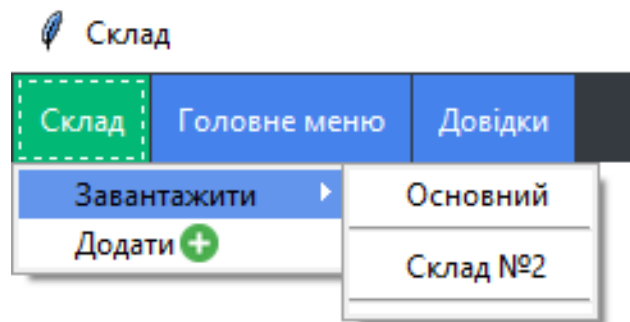


Рисунок Д.2 - Каскадне меню з можливістю завантаження і додавання складу

Склад

Склад Головне меню Довідки

## Основний Надходження товару

Номер ордеру 🔍


№	Номер ордеру	Продукт	Кількість	Склад	Зона зберігання	Цех	Номер партії	Дата	Сума
670	1	Хліб	5	Основний	Бакалея	1	1	2024-05-15	170


Рисунок Д.3 - Меню «Надходження товару»

Новий продаж


## Додавання продажу

Номер ордеру:

Замовник:  

Назва продукту:  

Кількість:

Дата:  

Сума:

Рисунок Д.4 - Вікно додавання продажу

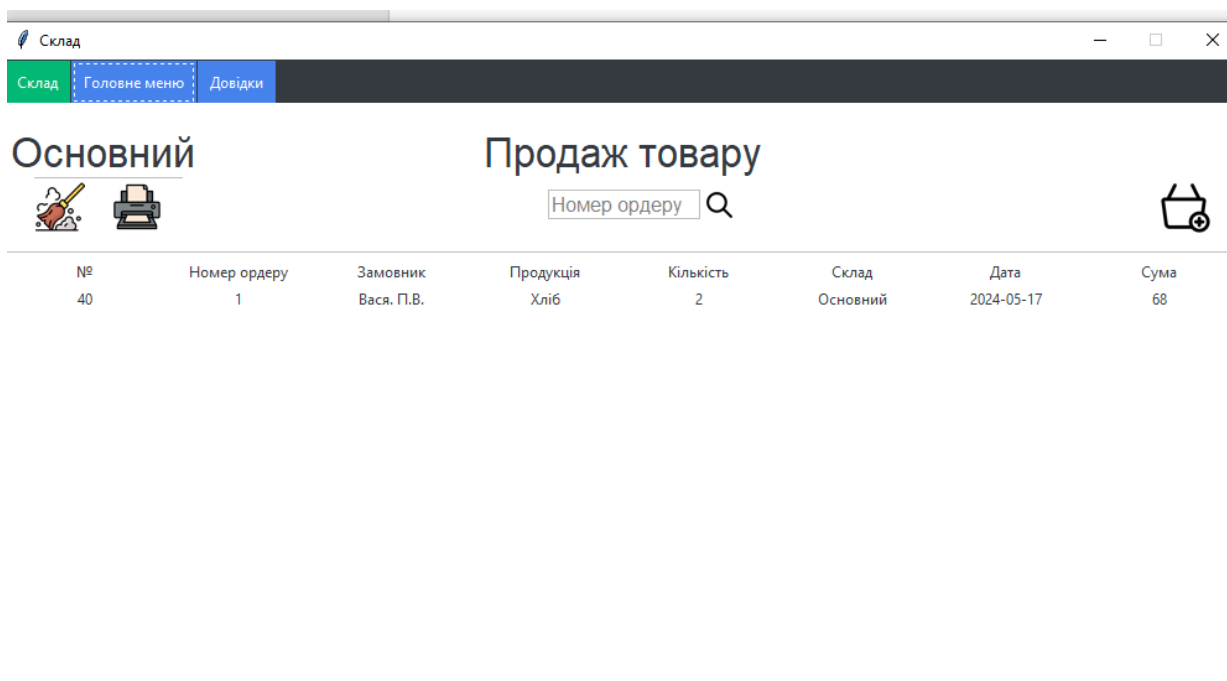


Рисунок Д.5 - Меню «Продаж»

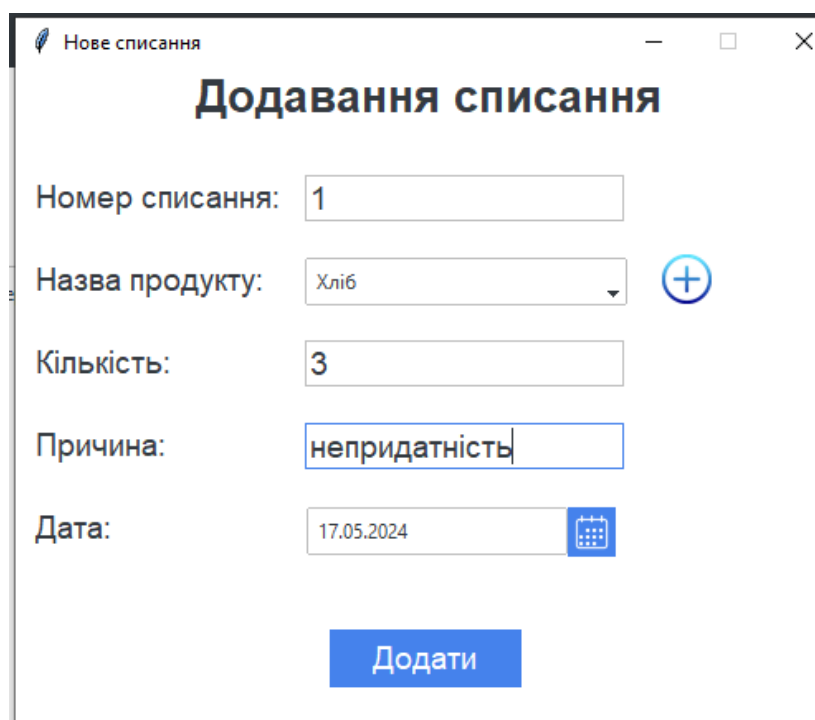


Рисунок Д.6 - Вікно додавання списання

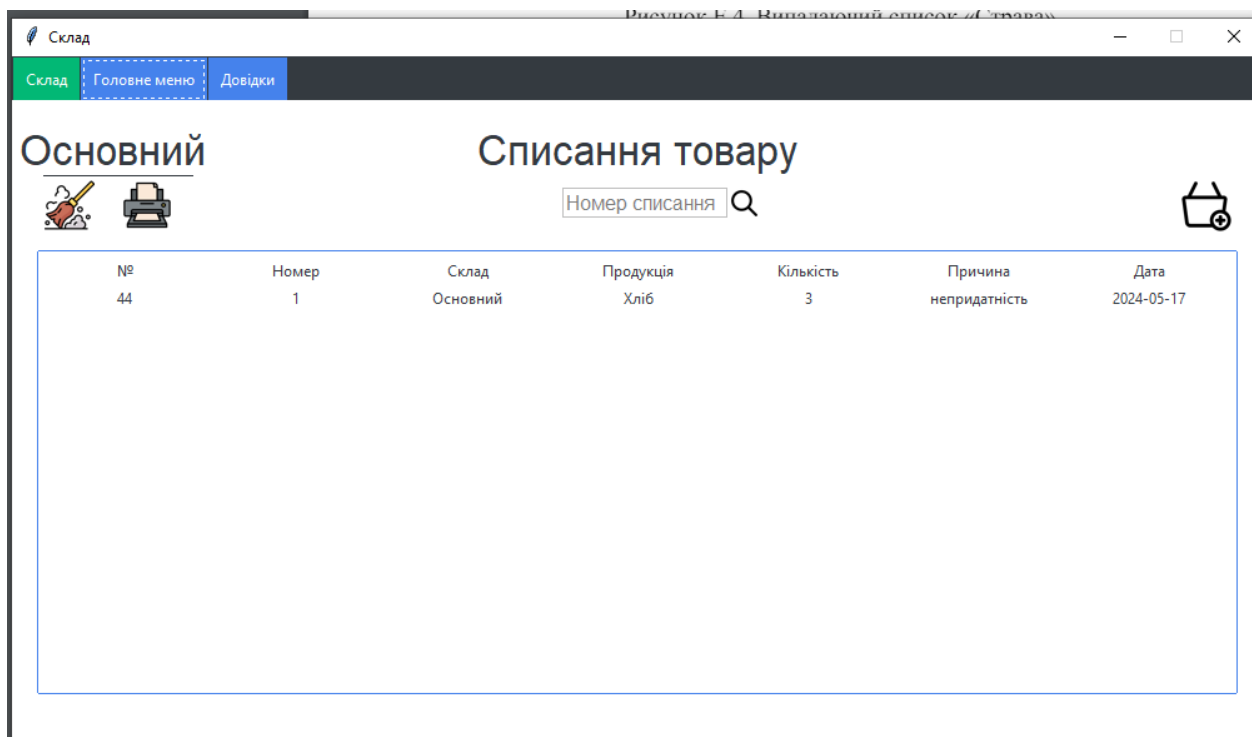


Рисунок Д.7 - Меню «Списання»

Новий замовник

### Додавання замовника

ПІБ замовника:

Телефон замовника:

Статус замовника:

Пошта:

Підприємство:

Адреса:

Реквізити:

Код ЄДРПОУ:

**Додати**

Рисунок Д.8 - Вікно додавання інформації про клієнта (замовника)

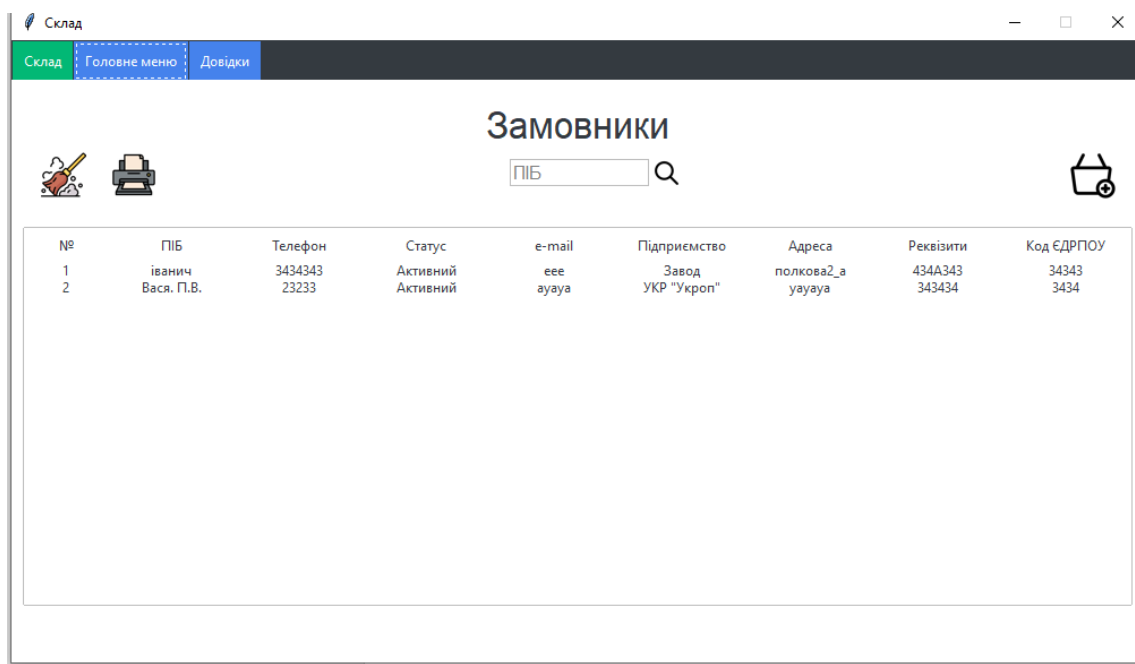


Рисунок Д.9 - Меню «Клієнти»

Нова ТТН

## Додавання ТТН

Номер ТТН:

Замовник:

Назва продукту:

Кількість:

Ціна/од:

Водій:

Транспорт:

Дата:

Сума:

**Додати**

Рисунок Д.10 - Вікно додавання ТТН

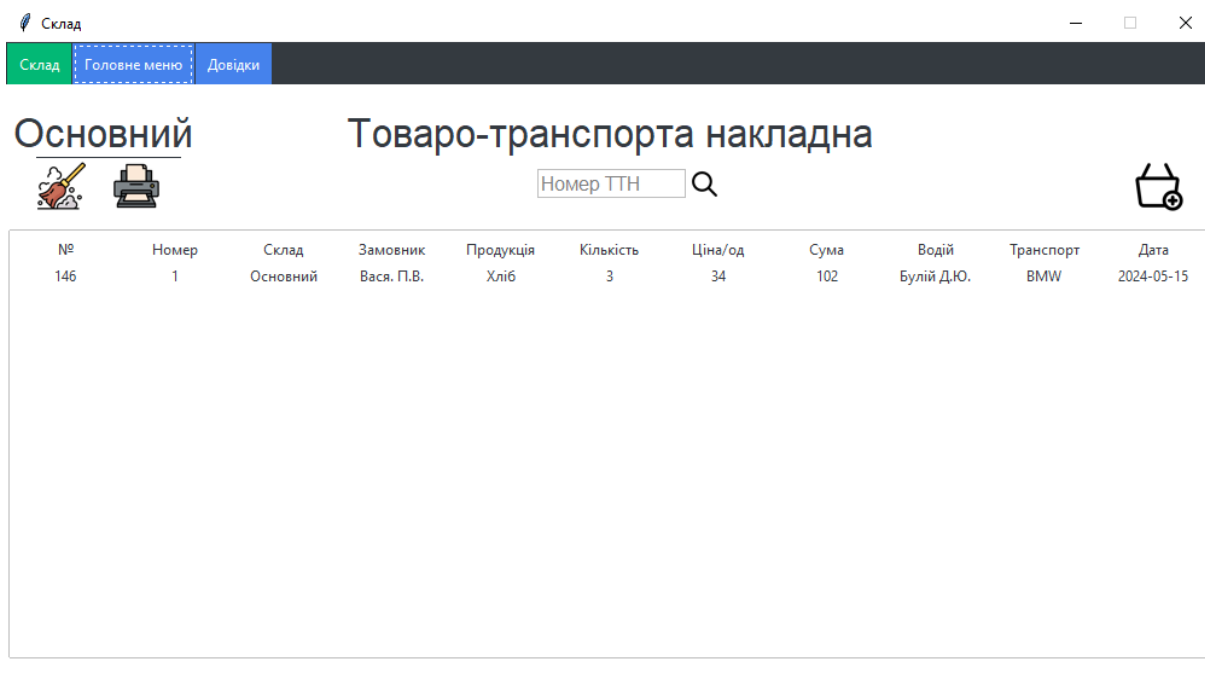


Рисунок Д.11 - Меню «ТТН»

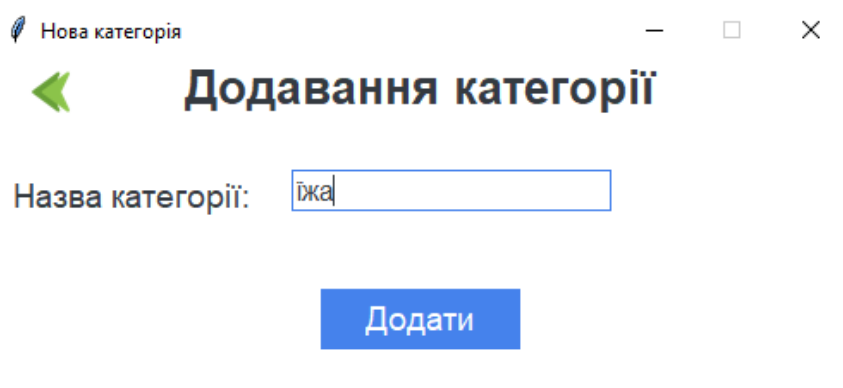


Рисунок Д.12 - Вікно додавання категорії товару

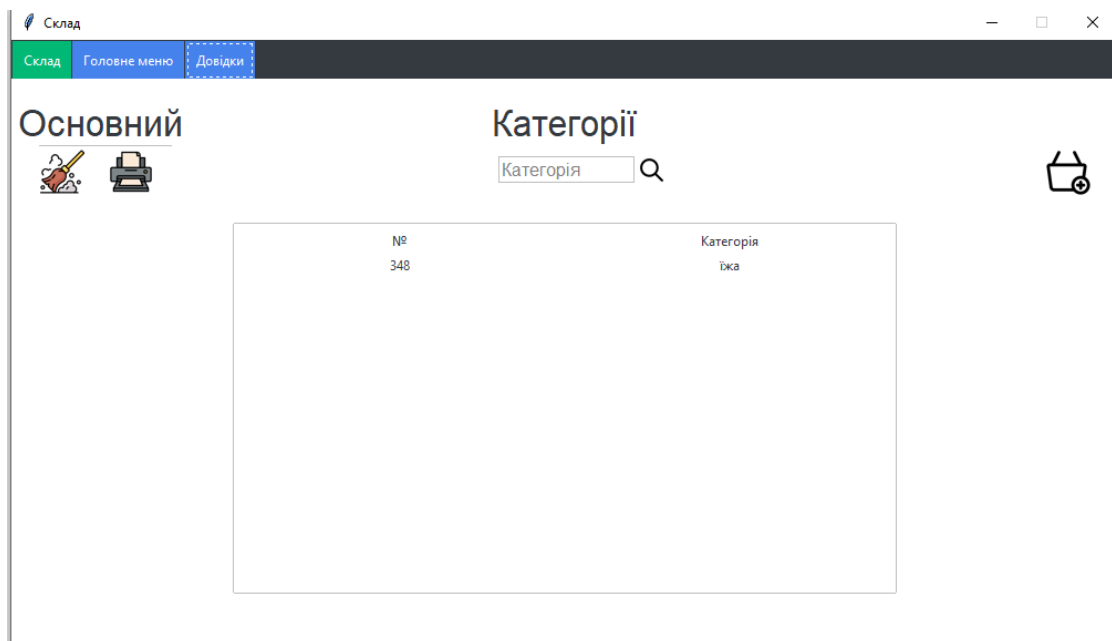


Рисунок Д.13 - Меню «Категорії»

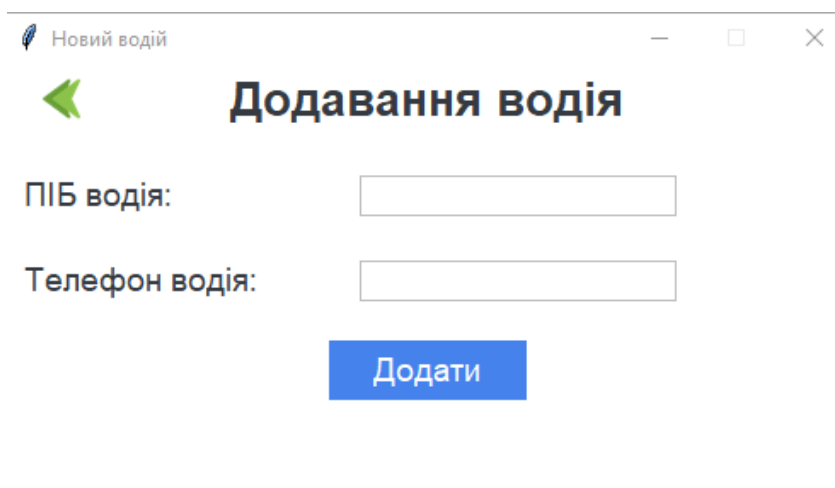


Рисунок Д.14 - Вікно додавання інформації про водія

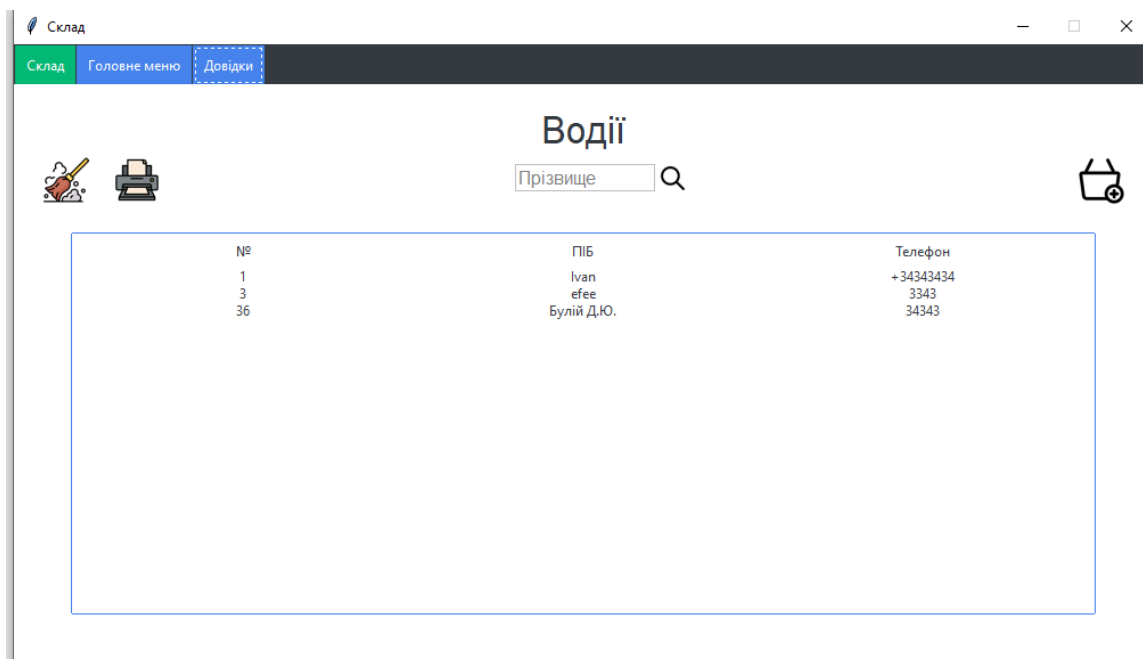


Рисунок Д.15 - Меню «Водій»

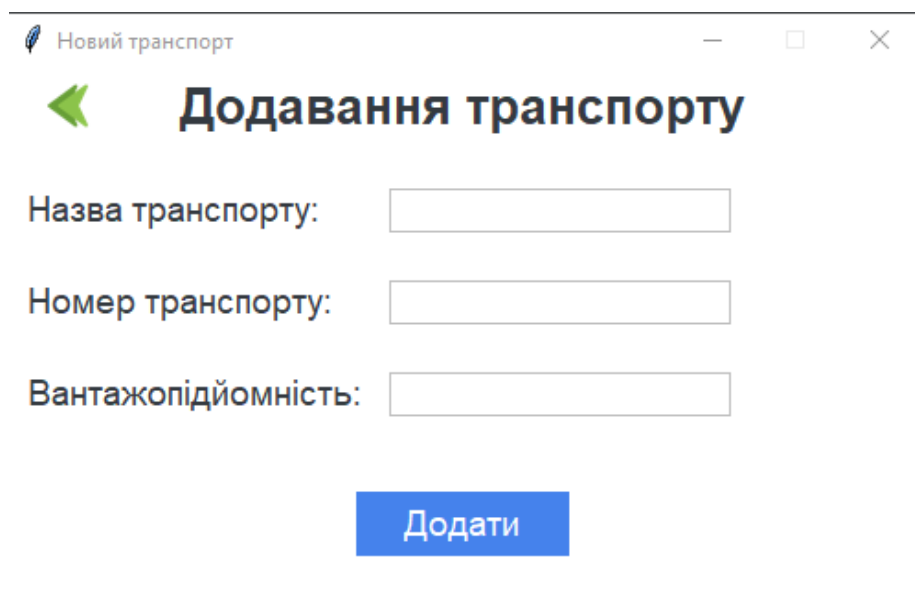


Рисунок Д.16 - Вікно додавання автотранспорту

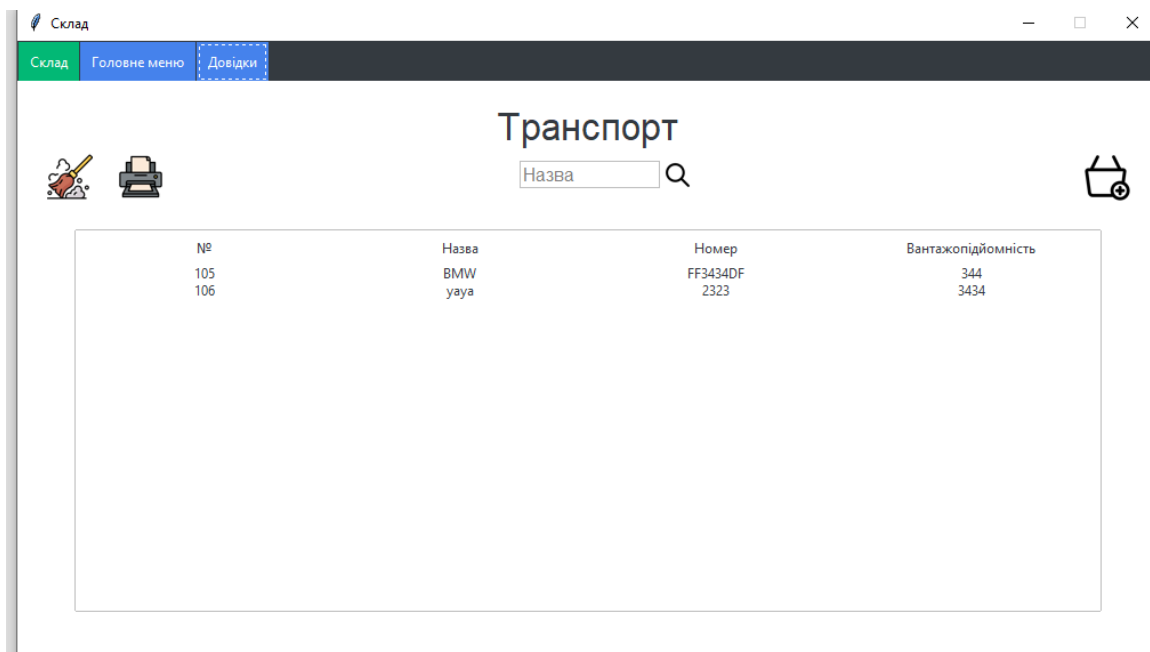


Рисунок Д.27 Меню «авто»

**ДОДАТОК Е. ФРАГМЕНТИ КОДУ**

Код підключення до бази даних:

```
import psycopg2

def execute_sql_query_insert(conn_str, sql_query, params=None):
    connection = psycopg2.connect(conn_str)
    cursor = connection.cursor()
    if params:
        cursor.execute(sql_query, params)
    else:
        cursor.execute(sql_query)
    connection.commit()
    cursor.close()
    connection.close()

def execute_sql_query_get(conn_str, sql_query, params=None):
    connection = psycopg2.connect(conn_str)
    cursor = connection.cursor()
    if params:
        cursor.execute(sql_query, params)
    else:
        cursor.execute(sql_query)

    result = cursor.fetchall()
    # Close the cursor and connection
    cursor.close()
    connection.close()
    return result # Return the query result

host = 'localhost'
port = '5432'
database_name = 'sclad'
```

```

username = 'postgres'
password = '25122002'
conn_str = f"host='{host}' port='{port}' dbname='{database_name}'
user='{username}' password='{password}'"

```

Код конструктора класу WarehouseAplic:

```

def __init__(self, login_form_instance):
    self.login_form_instance = login_form_instance
    self.new_window = tk.Toplevel(self.login_form_instance.root)
    self.new_window.title("Склад")
    self.new_window.configure(background="white")
    self.new_window.resizable(False, False)
    self.window_width = 1000
    self.window_height = 550
    self.center_window()
    self.frame = ttk.Frame(self.new_window, bootstyle="dark", width=1000,
height=35)
    self.frame.pack()
    self.frame_menu = tk.Frame(self.new_window, bg="White")
    self.empty_frame = tk.Frame(self.new_window, bg="White")
    self.zapas_menu = tk.Frame(self.new_window, bg="White")
    self.thona_frame = Frame(self.new_window, bg="White")
    self.categ_frame = Frame(self.new_window, bg="White")
    self.avto_frame = Frame(self.new_window, bg="White")
    self.driver_frame = Frame(self.new_window, bg="White")
    self.oreder_come_menu = tk.Frame(self.new_window, bg="White")
    self.spus_frame = Frame(self.new_window, bg="White")
    self.oreder_sell_menu = tk.Frame(self.new_window, bg="White")
    self.customer_frame = Frame(self.new_window, bg="White")
    self.ttn_frame = Frame(self.new_window, bg="White")
    self.menu_list()

```

```

self.sep = ttk.Separator(self.frame_menu, bootstyle="dark")
self.sep.pack(padx=400, pady=75, fill=ttk.X)
self.sklad_names = self.get_scl()
self.open_image()
self.check_and_show_menu()
self.label_text = None
self.chosen_name = None

```

Код додавання вікна складу:

```

def entry_data(root, form3_instance):
    labels_list = []
    def add():
        for label in labels_list:
            label.destroy()
        existing_names = [name[1] for name in get_scl()]
        name = entry_sk1.get()
        if name:
            if name in existing_names:
                label_error_add1 = ttk.Label(popup, text="Такий склад вже
існує", foreground="red", background="White")
                label_error_add1.place(x=35, y=120)
                labels_list.append(label_error_add1)
            else:
                sql_query = f"INSERT INTO Склад (Назва_складу) VALUES
(N'{name}')"
                execute_sql_query_insert(conn_str, sql_query)
                popup.withdraw() # Скрываем окно, но сохраняем его
состояние
                form3_instance.check_data()
                choose_manu(popup, form3_instance)
        else:

```

```
label_error_add = ttk.Label(popup, text="Введіть назву",
                             foreground="red", background="White")
label_error_add.place(x=60, y=120)
labels_list.append(label_error_add)

popup = tk.Toplevel(root)
popup.title("Додавання складу")
popup.geometry("200x150")

style = ttk.Style()
style.configure("White.TButton", background="white", borderwidth=0)

label = ttk.Label(popup, text="Введіть назву складу", font=("Arial", 12))
label.pack(pady=10)
entry_skl = Entry(popup, font=("Arial", 14), width=10)
entry_skl.pack(pady=5)

button_add = Button(popup, text="Додати", font=("Arial",
12), background='white', bd=0,
highlightthickness=0, highlightbackground='white', command=add)
button_add.pack(pady=7)

popup.resizable(False, False)

root_width = root.winfo_reqwidth()
root_height = root.winfo_reqheight()
popup_width = 300
popup_height = 140
x_coordinate = root.winfo_x() + (root_width - popup_width) // 2
y_coordinate = root.winfo_y() + 70 + (root_height - popup_height) // 2
```

```
popup.geometry(f"{popup_width}x{popup_height}+{x_coordinate}+{y_
coordinate}")
```

```
def on_close():
```

```
    popup.withdraw()
```

```
    choose_manu(root, form3_instance)
```

```
popup.protocol("WM_DELETE_WINDOW", on_close)
```

Код сортування даних в таблиці Treeview

```
def sort_treeview_column(tree, col, reverse=False):
```

```
    if col in ["Ціна за од", "Кількість палетів"]:
```

```
        data = [(float(tree.set(child, col)), child) for child in tree.get_children(")]
```

```
        data.sort(reverse=reverse)
```

```
        for index, (val, child) in enumerate(data):
```

```
            tree.move(child, "", index)
```

```
        tree.heading(col, command=lambda: sort_treeview_column(tree, col, not
reverse))
```

```
    else:
```

```
        data = [(tree.set(child, col).lower(), child) for child in tree.get_children(")]
```

```
        data.sort(reverse=reverse)
```

```
        for index, (val, child) in enumerate(data):
```

```
            tree.move(child, "", index)
```

```
        tree.heading(col, command=lambda: sort_treeview_column(tree, col, not
reverse))
```

```
for col in columns:
```

```
    tree.heading(col, text=col, command=lambda c=col:
```

```
sort_treeview_column(tree, c))
```

Код друку даних з таблиці treeview:

```
def print_to_pdf():
    pdf = FPDF()
    pdf.add_page()
    font_path =
r"C:\Users\Данил\PycharmProjects\ProjectX_Buliy\_fonts\DejaVuSansM
ono.ttf"
    pdf.add_font("DejaVuSansMono", fname=font_path, style="", uni=True)
    pdf.set_font("DejaVuSansMono", size=10)
    page_width = pdf.w
    title_text = "Запаси"

    title_width = pdf.get_string_width(title_text)
    # Вычисляем позицию для выравнивания по центру
    x_position = (page_width - title_width) / 2

    # Устанавливаем позицию X для центрирования заголовка
    pdf.set_x(x_position)
    pdf.set_font("DejaVuSansMono", size=16) # Устанавливаем жирный
    шрифт с размером 12
    pdf.cell(0, 10, title_text, 0, 1, 'C') # Выводим заголовок
    pdf.set_font("DejaVuSansMono", size=10)

    columns = ("№", "Назва", "Кількість", "Ціна/од", "Зберігання",
    "Склад", "Артикул", "Категорія")

    for col, text in enumerate(columns):
        pdf.cell(24, 7, text, border=1, ln=False)

    pdf.ln()
```

```
for row_id in tree.get_children():
    values = tree.item(row_id, "values")
    for value in values:
        pdf.cell(24, 7, str(value), border=1, ln=False)
    pdf.ln()

pdf.output("thapasu.pdf")
```