

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА


на тему:

«Інформаційна технологія автоматизації роботи супермаркетів»

Виконала: студентка 2-го курсу,
групи ІКН-21м, спеціальності
122 – Комп'ютерні науки


Буксир Я. О.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН

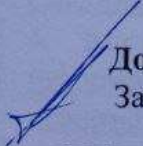

Крилик Л. В.
(прізвище та ініціали)

« 15 » 12 2022 р.

Опонент: к.т.н., доцент каф. АІТ


Богач І. В.
(прізвище та ініціали)

« 15 » 12 2022 р.


Допущено до захисту
Завідувач кафедри КН
д.т.н., проф. Яровий А.А.
(прізвище та ініціали)

« 16 » 12 2022 р.

Вінницький національний технічний університет
 Факультет інтелектуальних інформаційних технологій та автоматизації
 Кафедра комп'ютерних наук
 Рівень вищої освіти II-й (магістерський)
 Галузь знань – 12 Інформаційні технології
 Спеціальність – 122 Комп'ютерні науки
 Освітньо-професійна програма – Системи штучного інтелекту

ЗАТВЕРДЖУЮ

Директор ТОВ «Аксіома В»

Костинюк В. В.

« 15 »

(підпис)
2022 року



ЗАТВЕРДЖУЮ

Завідувач кафедри КН

д.т.н., проф. Яровий А.А.

« 15 » 09

(підпис)
2022 року

ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ

Буксир Яні Олександрівні
 (прізвище, ім'я, по батькові)

1. Тема роботи «Інформаційна технологія автоматизації роботи супермаркетів» керівник роботи к.т.н., доц., доц. каф. КН, Крилик Л. В. затверджені наказом вищого навчального закладу від «14» 09 2022 року №103
2. Строк подання студентом роботи 18 листопада 2022 року
3. Вихідні дані до роботи: мова програмування – об'єктно-орієнтована, максимальна кількість можливих категорій – 20 пунктів; максимальна кількість можливих підкатегорій – 20 пунктів; максимальна довжина назви категорії – 40 символів; максимальна довжина назви підкатегорії – 40 символів; тип інтерфейсу користувача – інтуїтивно зрозумілий; програмний додаток для автоматизації роботи супермаркетів, який реалізує завантаження товарів на сенсорні ваги самообслуговування.
4. Зміст текстової частини: вступ, обґрунтування доцільності розробки інформаційної технології автоматизації роботи супермаркетів, моделювання інформаційної технології автоматизації роботи супермаркетів, програмна реалізація інформаційної технології автоматизації роботи супермаркетів, економічна частина, висновки, список використаних джерел, додатки.
5. Перелік ілюстративного матеріалу (з точним зазначенням об'язкових креслень): UML-діаграма асоціацій класів інформаційної технології автоматизації роботи супермаркетів; схема ER-моделі бази даних інформаційної технології автоматизації роботи супермаркетів; схема зв'язків у базі даних інформаційної технології автоматизації роботи супермаркетів; структурна схема інформаційної технології автоматизації роботи

супермаркетів; схема загального алгоритму функціонування інформаційної технології; схема алгоритму функціонування модуля сортування товарів перед завантаженням на сенсорні ваги самообслуговування; діаграма створення товарів на емуляторі екрану вагів; діаграма створення файлу XML для завантаження інформації на ваги; загальний вигляд інтерфейсного вікна налаштувань для роботи з вагами; загальний вигляд інтерфейсного вікна емулятора екрану вагів після додавання зображення на товари.

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-3	Крилик Л.В., к.т.н., доц. каф. КН	<i>14.09.2022</i>	<i>14.09.2022</i>
4	Нікіфорова Л. О., к.е.н, доц. каф. ЕПВМ	<i>14.09.2022</i>	<i>14.09.2022</i>

7. Дата видачі завдання 14.09 2022 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Обґрунтування доцільності розробки	<i>14.09.2022 р - 01.10.2022 р</i>	Розділ 1
2	Моделювання інформаційної технології автоматизації роботи супермаркетів	<i>02.10.2022 р - 16.10.2022 р</i>	Розділ 2
3	Програмна реалізація інформаційної технології автоматизації роботи супермаркетів	<i>17.10.2022 р - 07.11.2022 р</i>	Розділ 3
4	Підготовка економічної частини	<i>08.11.2022 р - 21.11.2022 р</i>	Розділ 4
5	Апробація та/або впровадження результатів дослідження	<i>23.11.2022 р - 01.12.2022 р</i>	Тези доповіді, довідка про впровадження розробки
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	<i>02.12.2022 р - 14.12.2022 р</i>	Пояснювальна записка, графічний матеріал, презентація

Студентка

Керівник роботи

[Signature]

[Signature]

Буксир Я. О.

Крилик Л. В.

АНОТАЦІЯ

УДК 004.42

Буксир Я. О. Інформаційна технологія автоматизації роботи супермаркетів. Магістерська кваліфікаційна робота зі спеціальності 122 – Комп'ютерні науки, освітня програма – Системи штучного інтелекту. Вінниця: ВНТУ, 2022 – 128 с.

Укр. мовою. Бібліогр.: 45 назв; рис.: 22; табл. 8.

У магістерській кваліфікаційній роботі проведено аналіз предметної галузі автоматизації роботи супермаркетів і порівняння різноманітних методів та технологій призначених для автоматизації бізнесу.

Виконано аналіз аналогів, розглянуто методи та технології, які використовуються для вирішення подібних задач. Проведено обґрунтування вибору платформи та бази для програмного додатку автоматизації роботи супермаркетів. Було обрано BAS Управління торгівлею (8.3.16.1690). Розроблено алгоритм роботи програмного додатку та відповідне програмне забезпечення на вбудованій мові програмування 1С: Підприємство. Аналіз роботи програмного забезпечення показав підвищення зручності завантаження товарів на ваги та простіший вибір товарів для покупців.

Дана робота допомагає проаналізувати питання, які стосуються автоматизації роботи супермаркетів та допомагає актуально підібрати технологію, яка покращує роботу бізнесу.

В роботі приведені результати досліджень та аналіз роботи інформаційної технології автоматизації роботи супермаркетів, представленого на тестових прикладах та аналіз існуючих аналогів. Доведена економічна доцільність розробки.

Ілюстративна частина складається з 10 плакатів із результатами моделювання.

Ключові слова: автоматизація, бізнес, супермаркети, ваги самообслуговування, покупці, товари, інформаційна технологія.

ABSTRACT

Buksyr Y. O. Information technology of supermarket automation. Master's work in specialty 122 – Computer science. Vinnytsia: VNTU, 2022. 128 p.

In Ukrainian language Bibliography: 45 titles; Fig.: 22; table 8

In the master's qualification work, an analysis of the subject area of supermarket automation and a comparison of various methods and technologies designed for business automation were carried out.

An analysis of analogues was carried out, methods and technologies used to solve similar problems were considered. The justification of the choice of the platform and the base for the software application for the automation of supermarket operation was carried out. The BAS of the Trade Administration (8.3.16.1690) was chosen. The algorithm of the software application and the corresponding software in the built-in programming language 1C: Enterprise have been developed. Analysis of the software's performance showed an increase in the ease of loading goods onto the scales and an easier selection of goods for buyers.

This work helps to analyze the issues related to the automation of supermarket operations and helps to choose a technology that improves business performance. The work presents the results of research and analysis of the information technology of supermarket automation, presented on test examples and analysis of existing analogues. The economic feasibility of the development has been proven.

The illustrative part consists of 10 posters with simulation results.

Keywords: automation, business, supermarkets, self-service scales, buyers, goods, information technology.

ЗМІСТ

ВСТУП	6
1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АВТОМАТИЗАЦІЇ РОБОТИ СУПЕРМАРКЕТІВ	10
1.1 Аналіз технічних рішень програмного забезпечення для автоматизації роботи супермаркетів	13
1.2 Огляд програмних систем-аналогів	17
1.3 Формулювання вимог та постановка задачі	20
1.4 Висновок до розділу 1	22
2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АВТОМАТИЗАЦІЇ РОБОТИ СУПЕРМАРКЕТІВ	23
2.1 Обґрунтування вибору методу формування списку товарів	23
2.2 Розробка математичної моделі формування списку товарів на сенсорні ваги самообслуговування	27
2.3 Розробка UML-діаграми класів інформаційної технології автоматизації роботи супермаркетів	31
2.4 Розробка ER-моделі бази даних автоматизації роботи супермаркетів та обґрунтування вибору бази даних	33
2.5 Висновок до розділу 2	43
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АВТОМАТИЗАЦІЇ РОБОТИ СУПЕРМАРКЕТІВ	45
3.1 Структурна організація інформаційної технології автоматизації роботи супермаркетів	45
3.2 Розробка схеми алгоритму роботи інформаційної технології автоматизації роботи супермаркетів	48
3.3 Обґрунтування вибору мови та середовища програмування	53
3.4 Опис програмних рішень інформаційної технології автоматизації роботи супермаркетів	57

3.5 Тестування роботи програми та аналіз результатів	65
3.6 Висновок до розділу 3	70
4 ЕКОНОМІЧНА ЧАСТИНА	72
4.1 Комерційний та технологічний аудит науково-технічної розробки.....	72
4.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи	76
4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором	81
4.4 Висновок до розділу 4	87
ВИСНОВКИ.....	88
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	90
Додаток А (обов'язковий) Результат перевірки на плагіат в онлайн-системі UNICHECK	95
Додаток Б (обов'язковий) Лістинг програми	96
Додаток В (обов'язковий) Ілюстративна частина.....	Ошибка! Закладка не определена.
Додаток Г (довідниковий) Інструкція користувача	115
Додаток Д (довідниковий) Довідка про участь в конкурсі студентських наукових робіт	124
Додаток Е (довідниковий) Довідка про впровадження розробки	125

ВСТУП

Актуальність теми. Мережі супермаркетів зростають з кожним днем та завойовують все більшої довіри в покупців. Зі зростанням попиту покращується маркетинг, реклама, обслуговування та внутрішні процеси в роботі супермаркетів. Супермаркети збільшують обсяги товарів, що привертає увагу покупців та збільшує кількість продажів. Але разом з цим виникають проблеми в магазинах з невчасною реалізацією товару, великою кількістю зіпсованої продукції. Крім того, в таких випадках часто виникає проблема в неактуальності інформації про товар, що засмучує покупця. Через некоректну інформацію про товар, покупці часто відмовляються від товару після зважування його або ж навіть на касі.

Супермаркети надають великий перелік товарів, дають можливість самому обрати потрібний товар, зважити його і навіть просканувати та провести оплату на касах самообслуговування. Досить популярним є використання сенсорних вагів самообслуговування, що використовують як в малих супермаркетах, так і у великих мережах. І хоча нині це найкраще рішення для зважування вагового товару, але і в адміністрації супермаркетів і в покупців досі виникають проблеми в роботі з ними. Адміністрація супермаркетів часто має проблеми з налаштуванням, чи завантаженням товарів на ваги. А покупці в свою чергу часто не можуть знайти їм потрібні товари на вагах або ж роздрукувати етикетку [1].

Саме тому розробка та застосування інформаційної технології автоматизації роботи супермаркетів, яка за рахунок автоматизації поточних операцій (налаштування та завантаження товарів на сенсорні ваги самообслуговування) спрощує роботу персоналу та збільшує ефективність роботи супермаркетів і магазинів є **актуальною**.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська кваліфікаційна робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Розробка прикладних інтелектуальних інформаційних технологій та систем» і плану наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження. Метою магістерської кваліфікаційної роботи є розширення функціональних можливостей інформаційної технології автоматизації роботи супермаркетів, що дозволить підвищити ефективність та швидкодію завантаження товарів на сенсорні ваги самообслуговування.

Для досягнення поставленої мети необхідно розв'язати такі **завдання**:

- провести аналіз та обґрунтувати доцільність розробки інформаційної технології автоматизації роботи супермаркетів;
- провести аналіз та виконати вибір методів і технологій для автоматизації роботи супермаркетів;
- розробити математичну модель формування списку товарів на сенсорні ваги самообслуговування;
- обґрунтувати вибір алгоритму реалізації функції сортування та завантаження товарів на сенсорні ваги самообслуговування;
- розробити структурну організацію інформаційної технології автоматизації роботи супермаркетів;
- провести тестування програми та проаналізувати отримані результати;
- економічно обґрунтувати доцільність розробки інформаційної технології автоматизації роботи супермаркетів.

Об'єкт дослідження – процес автоматизації роботи супермаркетів.

Предмет дослідження – методи та програмні засоби, що надають змогу реалізовувати та переглядати програмний засіб та результат роботи.

Методи дослідження. У роботі використані такі методи наукових досліджень: аналіз інформаційної технології автоматизації роботи супермаркетів, методи обробки цифрової інформації, методи кластеризації,

методи взаємодії між серверами, методи комунікації з базою даних, методи взаємодії між сторонніми плагінами, методи створення робочої панелі для адміністратора супермаркету, методи об'єктно-орієнтованого програмування для отримання подальших результатів роботи програми.

Наукова новизна одержаних результатів полягає в наступному: вдосконалено інформаційну технологію автоматизації роботи супермаркетів, яка відрізняється від аналогів використанням математичної моделі на основі методу ранжування для формування списку товарів, надання користувачеві нових функцій та збільшення швидкодії процесу обміну даними між сенсорними вагами самообслуговування та програмою. Це збільшує ефективність роботи персоналу супермаркету та робить зручнішим процес купівлі товарів для покупців.

Практичне значення одержаних результатів полягає у наступному:

1. Удосконалено алгоритм функціонування модуля управління товарами та їх сортування.
2. Здійснено програмну реалізацію інформаційної технології автоматизації роботи супермаркетів.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, чітким та лаконічним виведенням аналітичних співвідношень, порівнянням результатів з відомими та збіжністю результатів математичного моделювання з результатами, що отримані під час впровадження розроблених програмних засобів.

Особистий внесок магістранта. Результати магістерської кваліфікаційної роботи отримані самостійно. В публікації у співавторстві здобувачу належить дослідження перспектив розробки інформаційної технології автоматизації роботи супермаркетів [1, 2].

Апробація результатів магістерської кваліфікаційної роботи.

Результати досліджень було апробовано на LI науково-технічній конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2022) м. Вінниці у 2022 р. [2], а також в I турі Всеукраїнського конкурсу студентських наукових робіт у ВНТУ 2021/2022 н.р. За підсумками I-го туру наукова робота була рекомендована для участі у II-му турі конкурсу і направлена до Центральноукраїнського національного технічного університету, м. Кропивницький 2022 р.

Публікації магістерської кваліфікаційної роботи. За результатами магістерської кваліфікаційної роботи опубліковано тези доповідей на науково-технічних конференціях [1, 2] та отримано свідоцтво про реєстрацію авторського права на твір у формі комп'ютерної програми – «Інформаційна технологія автоматизації роботи супермаркетів» [3].

Результати роботи пройшли експериментальне випробування та впровадженні для подальшого використання ТОВ «Аксиома В» м. Вінниця.

1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АВТОМАТИЗАЦІЇ РОБОТИ СУПЕРМАРКЕТІВ

Супермаркет – різновид універсального магазину самообслуговування з торговою площею від 400 м² до 2499 м², у якому покупцю пропонується не менше двох тисяч найменувань продовольчих товарів, широкий асортимент непродовольчих товарів, а також пропонуються додаткові послуги населенню, зокрема – стоянка автотранспорту, послуги з харчування, комплектування подарункових наборів тощо [4].

Основною метою супермаркету є задоволення особистих потреб споживача. Пропонуючи широкий спектр товарів першої необхідності, супермаркети є відмінною альтернативою стихійним ринкам та спеціалізованим магазинам. Через ширший асортимент товарів народного споживання супермаркети в багатьох випадках витісняють звичайні продуктові магазини.

Існують такі переваги робити покупки саме в супермаркеті:

- низькі ціни;
- акції та знижки;
- парковки;
- зручні візочки, що дозволяють не носити продукти в руках;
- покупець почувається більш вільно і зручно, адже може вибирати без продавця і витратити стільки часу, скільки йому завгодно;
- правильне розташування товару стимулює до покупки;
- збільшення середнього чека, тому що покупець бачить більшу кількість товару та може підібрати додаткові;
- велика пропускна спроможність магазину;
- менша кількість працівників та водночас менше витрат на заробітну плату;
- відсутність натовпів у магазині;

– у покупців набагато знижується контакт із персоналом, що для певних груп населення важливо.

З автоматизацією роботи супермаркетів все більшого попиту набувають і сенсорні ваги самообслуговування, що є досить зручним та вже невід'ємним атрибутом супермаркету. Споживачам до вподоби самим контролювати вагу потрібного товару та ціну. Таким чином до супермаркету зростає довіра, оскільки споживач не переймається, що його можуть обважити.

Ваги самообслуговування застосовуються у відділах супер- і гіпермаркетів з продажу фруктів, овочів, сипучих продуктів, солодоців та кондитерських виробів. Застосування ваг самообслуговування дозволяє ритейлеру перекласти функцію зважування на покупця, тим самим заощаджуючи на персоналі торгового залу і підвищуючи лояльність [5].

Сенсорні ваги самообслуговування – нове покоління комп'ютеризованих ваг, з розширеними можливостями. Кількість товарів для програмування на вагах необмежена. Додаються нові функції – можливість проведення рекламних активностей, друку рецептів і купонів, інтеграція з електронними цінниками. Адміністрування ваг особливо зручно завдяки можливості віддаленого адміністрування та управління [5].

Але з появою таких вагів, з'являються і нові проблеми. При застосуванні таких ваг потрібно більше простору та квадратури, обов'язкове замовлення та купівля спеціального обладнання, навчання співробітників, розробка систем спостереження за можливими крадіжками, які за статистикою можуть збільшуватися за поганого захисту, збільшення навантажень на каси та формування черг. Зі сторони адміністрації виникають проблеми у завантаженні товарів на них, а зі сторони користувача виникає проблема в пошуку потрібного товару серед інших. У супермаркетах часто можна зустріти покупців, особливо похилого віку, що не можуть знайти потрібний товар на вагах. Такі проблеми псують репутацію та схильність покупців до супермаркету. Всі недоліки можна згладити, якщо правильно організувати процес, але на все це потрібно виділяти

окремі витрати. Загалом переваг більше, хоча багато залежить від розташування магазину, товару, який продається та інших дрібних нюансів.

Проблема завантаження товарів на ваги може бути пов'язана з вагами різних марок, різними базами чи конфігураціями. Адміністратори часто не можуть замінити стандартні картинки товарів, які вже є в базі або ж відредагувати послідовність товарів на вагах. Зазвичай дані передаються на ваги шляхом кодування їх в XML. Такий файл XML може містити у собі товари, ціни, зображення, тощо.

XML (розширювана мова розмітки) – це мова програмування, яка складається з оголошень у вигляді інформації і визначальних тегів. З її допомогою зручно зберігати і передавати будь-які дані. Мова не залежить від операційної системи і середовища обробки. XML служить для представлення деяких даних у вигляді структури, яку ви можете самі розробити або налаштувати під програму або сервіс. Саме тому таку мову називають розширюваною, і в цьому її головна перевага, за яку її так цінують [6].

Переваги мови XML:

- легкість читання та подача у простій формі;
- можливість створення різних структур;
- можливість обміну даними між різними пристроями чи платформами;
- популярність в різних сферах програмування;
- стандартний вигляд кодування.

Для коректного XML документа мають виконуватися такі умови:

- правильне оформлення документа;
- дотримано всіх синтаксичних правил XML;
- документ має відповідати семантичним правилам мови (які зазвичай задані в схемі XML або DTD (Document Type Definition)).

1.1 Аналіз технічних рішень програмного забезпечення для автоматизації роботи супермаркетів

Зі зростанням кількості супермаркетів зростає і потреба в якісному програмному забезпеченні, що допоможе зручно вести бізнес та автоматизувати їхню роботу. Звичайно кожен приватний підприємець, хоче максимально компактне та зрозуміле програмне забезпечення, що допоможе вести бізнес в одній програмі, не шукаючи різну інформацію то різним додаткам.

При виборі програмного забезпечення дуже важливо враховувати специфіку обліку в Україні та комплексність інформаційних систем управління діяльністю багатопрофільних підприємств.

Однією із найбільш популярних програм для ведення бізнесу є BAS, що за своєю суттю є продовженням лінійки програм «1С» для українського ринку [7]. Всі компанії різні, відрізняються структурою, розміром і видом діяльності, однак, завдяки широкій лінійці рішень BAS, кожен підприємець зможе підібрати рішення, яке ідеально підходить під специфіку і завдання конкретного підприємства. Особливістю системи є те, що вона в основному експлуатується на підприємствах України і має прийнятну для свого ринку вартість, європейські системи такого ж класу як правило коштують у декілька разів дорожче. Незважаючи на невисоку вартість, за всіма параметрами програма має конкурентоспроможну функціональність і переваги за рахунок наявності в системі модулів бухгалтерського і податкового обліку, які розроблені для всіх країн в яких вона використовується.

BAS (Business Automation Software) – комплекс прикладних рішень для автоматизації бізнес-процесів, що створені на основі програмної платформи BAF (Business Automation Framework).

Лінійка BAS у своєму арсеналі має як типові конфігурації, що є універсальними для будь-яких комерційних організацій, так і галузеві, створені

з урахуванням індивідуальної специфіки діяльності вузькопрофільних бізнесів [8].

Business Automation Software представляє собою продукт, сучасне рішення, яке допомагає автоматизувати всі процеси на українських підприємствах різних форм власності та сфер діяльності. Використовується для цього платформа BAF (Business Automation Framework). Це абсолютно нові програми, які розроблені виключно для українського ринку. Вони являють собою заміну вже застарілого софту. Зараз плавно здійснюється перехід з 1С: Підприємство на BAS [9].

BAS має вже досить велику кількість розроблених рішень, але найбільш популярні такі різновиди:

BAS ERP – рішення щодо впровадження комплексної інформаційної системи управління виробництва на великих підприємствах. Особливу увагу під час розробки було приділено реалізації функціональних можливостей, затребуваних великими підприємствами різних напрямків діяльності, зокрема з технічно складним виробництвом;

BAS Бухгалтерія ПРОФ, використовується для ведення бухгалтерського і податкового обліку українських компаній. Ви можете підтримувати декілька організацій на одній основі одночасно, працювати з торговими операціями, запасами, виробництвом, грошовими коштами і взаєморозрахунками з контрагентами, а також ви можете розраховувати податок на додану вартість, податок на прибуток та інші податки і заробітну плату;

BAS Управління торгівлею – це найкраще рішення, коли вам потрібно автоматизувати процеси вашого бізнесу в сфері торгівлі. Також підходить для роздрібною торгівлі. Оптимізується управління продажами товарів і послуг. Рішення дозволяє збільшити продуктивність праці багатьох служб підприємства, працювати з оперативною інформацією, яка відображає поточний стан операційної діяльності, швидко і в зручній формі отримувати звіти для прийняття рішень на різних рівнях;

BAS Роздрібна торгівля забезпечить автоматизацію роботи в торгових точках (магазини, мережеві магазини). Розташування касира також буде автоматизовано, що теж є перевагою. Рішення дозволяє автоматизувати облік товарних запасів на складах магазинів та облік грошових коштів в касах, а також може використовуватися в якості робочого місця касира. Значно спроститься робота з торговим обладнанням;

BAS Документообіг КОРП передбачає основну функцію управління бізнес-процесами та додатково розробники зробили акцент на веденні звітностей, документообігу в електронному вигляді. Крім того, програма допомагає налагодити спільну роботу співробітників, дає можливість працювати з будь-якими типами документів, організувати різні заходи та складати плани на кожен день. У такій системі набагато зручніше шукати документи;

BAS Управління холдингом – це програмне забезпечення підходить, якщо вам потрібно керувати бізнесом у великих масштабах, а також повністю або частково централізувати управління фінансами, закупівель та звітності;

BAS Комплексне управління підприємством – являє собою комплексне рішення з обліку і управління організацією, створення для неї єдиного інформаційного простору. Гарно підходить для малого і середнього бізнесу. Також передбачено управління грошовими коштами компанії, бюджетом, проведення моніторингу, спостереження за всіма показниками роботи підприємства. Проводиться регламентований облік;

BAS Медицина. Лікарня і BAS Медицина. Поліклініка – це 2 різновиди програмного забезпечення BAS, яке дає можливість створити спільний інформаційний простір в медичному закладі, а також автоматизувати трудомісткі і рутинні завдання. Обидві програми дають можливість управляти організацією, аптекою при лікарні, автоматизувати обіг медикаментів і витратних матеріалів. Можлива інтеграція з іншими програмами, які необхідні для ведення податкового і бухгалтерського обліку [9].

Для автоматизації, управління, контролю над операціями різного виду діяльності та ведення оперативного обліку супермаркету найбільше за своїм функціоналом підходить саме BAS Управління торгівлею. Продукт автоматизує такі задачі:

- планування та план-фактичний аналіз продажів, закупівель, збирання (розбирання);
- управління продажами, поставками, складськими запасами;
- управління відносинами з постачальниками та замовниками;
- управління замовленнями;
- самообслуговування клієнтів через web-сервіси;
- робота торгових представників та контрагентів;
- обробка претензій та зауважень;
- управління фінансами;
- облік та аналіз комерційних витрат та прибутку;
- управління взаєморозрахунками;
- аналіз цін і управління ціновою політикою, знижками та акціями;
- інтеграція з торговим обладнанням;
- аналітична звітність з торгової діяльності.

Ця програма має два режими роботи BAS та конфігуратор. Перший режим роботи призначений для користувача, а конфігуратор призначений для програміста чи адміністратора. Цей режим надає інструменти, необхідні для модифікації існуючої або створення нової конфігурації. Дерево об'єктів конфігурації – основний інструмент, що містить в собі практично всю інформацію з якої складається програма. До об'єктів дерева конфігурації відносять довідники, документи, реєстри, звіти, обробки тощо.

У свою чергу для того, щоб не вмішуватись в роботу цієї програми, використовують зовнішні програмні додатки, що в BAS називають зовнішньою обробкою.

1.2 Огляд програмних систем-аналогів

Оскільки проблема завантаження товарів існує вже тривалий час, було розроблено велику кількість програмних додатків для їх вирішення. Розглянемо деякі з вже існуючих програмних додатків та проведемо аналіз для вибору найбільш прийняттого, який використаємо як прототип.

Завантаження товарів на ваги Штрих-ПРИНТ 4.5 – це програмний додаток для завантаження товарів на ваги. Програмний додаток дає можливість встановити відбір за товарами, а саме натиснувши на кнопку «Відбір», відкриється вікно з налаштуваннями після встановлення яких відбудеться автоматичне заповнення списку товарів. Є можливість встановити характеристику ваговий, що поділить товари на вагові та штучні. Можна обрати вид ціни для товарів та можливість додати термін придатності на етикетку. Заповнити список товарів можна через кнопку «Добавити» або «Заповнити по відбору» (рис. 1.1) [10].

Недоліки програмного забезпечення:

- відсутня можливість перегляду загального вигляду товарів на вагах;
- відсутня можливість поділити товари на категорії (наприклад: овочі та фрукти);
- відсутня можливість завантажувати зображення товарів.

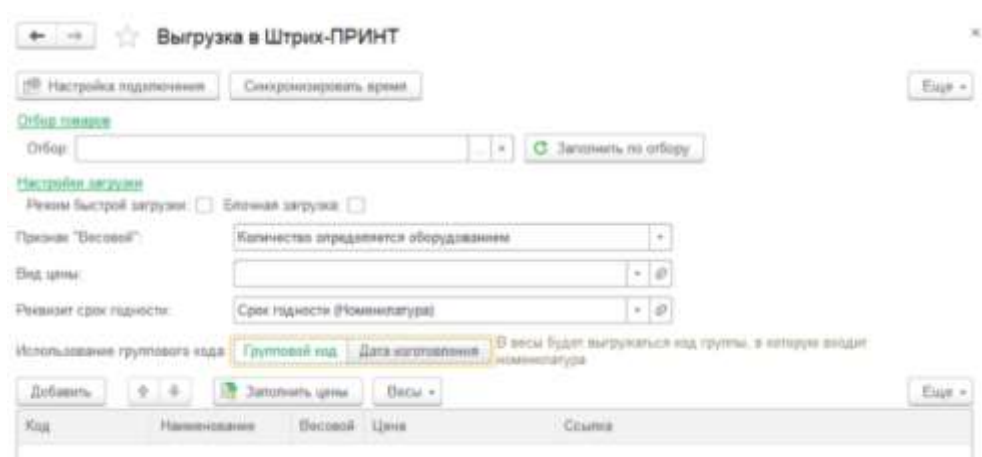


Рисунок 1.1 – Загальний вигляд інтерфейсного вікна програми Штрих-ПРИНТ

Цей програмний модуль є платний, що зменшує його популярність. Програмний модуль використовує типовий функціонал «Правила обміну з підключеним обладнанням offline». Дозволяє робити завантаження на ваги такої інформації: SKU номенклатури, найменування номенклатури, ціна номенклатури, PLU номенклатури, опис номенклатури (тільки для файлової версії), картинку номенклатури (тільки для файлової версії) (рис. 1.2) [11].

Недоліки:

- відсутня можливість переглянути, як будуть виглядати товари на вагах;
- відсутня можливість поділити товари на категорії;
- відсутня можливість замінити стандартні зображення товарів;
- вартість 30 000,00 грн.

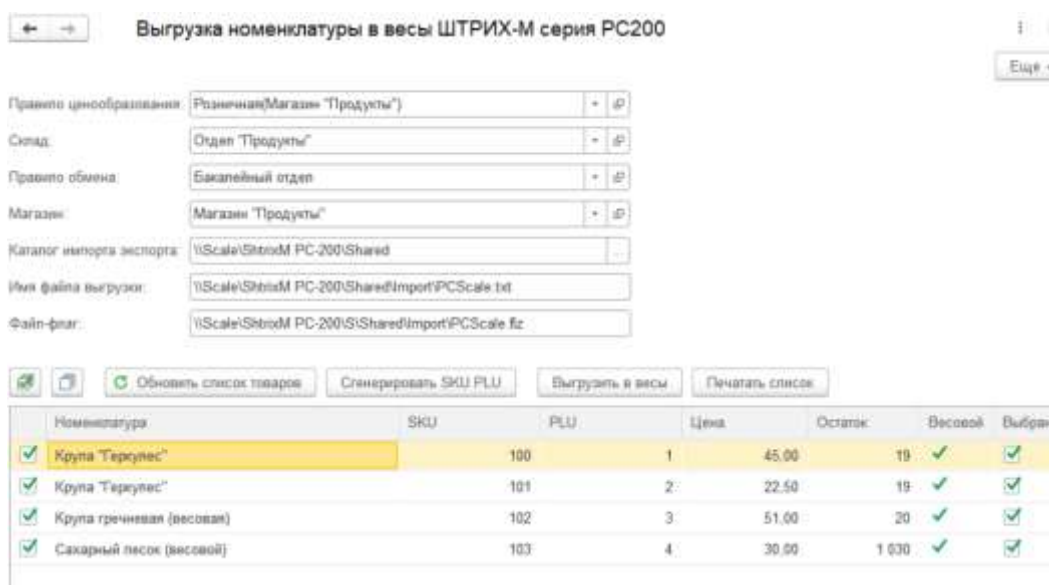


Рисунок 1.2 – Загальний вигляд інтерфейсного вікна програми ШТРИХ-М серії PC-200

Перед роботою з таким програмним модулем попередньо необхідно створити правило обміну (рис. 1.3). Правило обміну представляє собою, дані про підключене обладнання, магазин, склад та список товарів для завантаження.

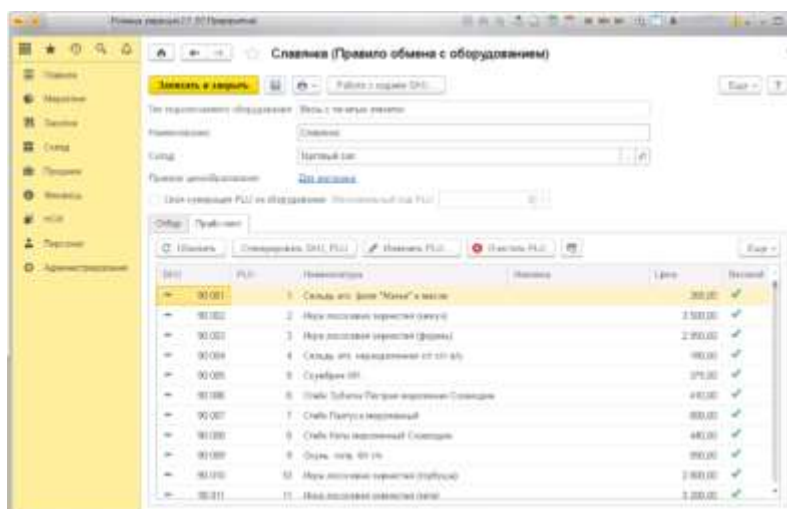


Рисунок 1.3 – Загальний вигляд інтерфейсного вікна програми для вагів Mettler toledo

Після створення правила обміну потрібно обрати тип вагів та паку на ПК для збереження налаштувань (рис. 1.4) [12]. Такий програмний додаток дозволяє завантажувати на ваги номенклатуру, ціни, штрихкоди, картинки, артикул товару.

Недоліки:

- відсутня можливість поділити товари на категорії;
- відсутня можливість змінити зображення;
- відсутня можливість переглянути зовнішній вигляд товарів на вагах, перед їх завантаженням;
- не зрозумілий інтерфейс.



Рисунок 1.4 – Загальний вигляд інтерфейсного вікна програми Mettler toledo

Проаналізувавши ці додатки, можна зробити висновок, що всі вони мають схожий функціонал, що допомагає адміністратору у завантаженні товарів на ваги. Але вони мають певні недоліки, а саме у всіх них проблема з поділом товарів на категорії та із заміною стандартних картинок. Серед проаналізованих програмних додатків найкращий функціонал має програмний модуль обробки обслуговування вагів ШТРИХ-М серії РС-200, який слугуватиме прототипом.

1.3 Формулювання вимог та постановка задачі

Задача цієї роботи полягає у розробці інформаційної технології автоматизації роботи супермаркетів.

Вимоги до клієнта:

- головна сторінка з налаштуваннями;
 - таблиця в якій є список вагів з якими можна зробити обміни;
 - таблиця товарів та їх даних, які будуть завантажуватись на ваги;
 - можливість обрати потрібну послідовність товарів (за алфавітом, за популярністю, за кількістю днів, що товар не продавався та за випадковою послідовністю);
 - сторінка з емулятором екрану вагів самообслуговування;
 - можливість створити потрібну кількість груп товарів;
 - можливість змінити товари, які належать до певної групи;
 - можливість змінити картинку групи товарів;
 - можливість змінити матричну сітку;
 - можливість змінити картинку товару;
 - можливість переглянути всі сторінки відображення груп та товарів, що належать до них.
- завершення роботи (вихід);

Вимоги до менеджера:

- можливість зберегти зроблені налаштування;

- можливість додавання та видалення інформації (товари, картинки, налаштування, ваги)
- можливість завантажити зроблені налаштування на всі підключені ваги в мережі або на одні з них;

Вимоги до сервера:

- зберігання даних користувача через зв'язок з базою даних;
- видача необхідних даних за запитом від клієнта.

Загальний опис функціоналу інформаційної технології:

Час від часу на вагах самообслуговування має оновлюватись інформація для того, щоб покупець завжди мав актуальний список товарів, їхню ціну та можливу знижку або акцію. Менеджеру, представлена можливість контролювати та редагувати обмін інформацією між програмним засобом в якому працює супермаркет та сенсорними вагами самообслуговування, а саме:

- представлена можливість змінювати групи товарів та їх вміст;
- змінювати картинки та стилізувати фон, групи та товари;
- перевіряти вигляд товарів до завантаження їх на ваги;
- відсортувати товари за потрібними показниками, наприклад:
 - якщо потрібно продати товар, який дуже довго стоїть на складі;
 - якщо потрібно продати товар на який є знижка чи акція;
 - якщо потрібно продати сезонний товар.

Потім менеджер може сам натиснути кнопку «Оновити» та завантажити всі зроблені налаштування на ваги, або ж обрати графік за яким автоматично буде відбуватись оновлення.

Отже, потрібно розробити інформаційну технологію, в якій буде реалізовано алгоритм формування списку товарів та завантаження їх на сенсорні ваги самообслуговування.

1.4 Висновок до розділу 1

У першому розділі було здійснено аналіз переваг супермаркетів серед інших торгових точок та актуальність використання в них сенсорних вагів самообслуговування. Було здійснено аналіз технічних рішень програмних додатків для автоматизації роботи супермаркетів. Для роботи обрано платформу BAS Управління торгівлею (8.3.16.1690), яка нині є доцільною для роботи супермаркету. Серед переваг є: збільшення продуктивності праці всіх служб торговельного підприємства; можливість працювати з оперативною інформацією, яка відображає поточний стан підприємства на даний час; швидко та у зручній формі отримувати звіти для прийняття рішень на різних рівнях.

Проведено огляд систем-аналогів, які створені для автоматизації завантаження товарів на сенсорні ваги самообслуговування, коротко охарактеризовано їх головні функції та наведено основні переваги і недоліки. Здійснено постановку задачі та опис призначення розроблюваної інформаційної технології, в результаті чого визначено задачі для подальшого дослідження.

Оскільки ця сфера є актуальною, то доцільно створювати інформаційну технологію, яка орієнтована як на користувача, тобто адміністрацію супермаркету, так і на споживача, покупців супермаркету.

2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АВТОМАТИЗАЦІЇ РОБОТИ СУПЕРМАРКЕТІВ

2.1 Обґрунтування вибору методу формування списку товарів

Правильне розміщення товарів на полицях вирішує відразу декілька завдань: привертає увагу покупця, підвищує лояльність, сприяє збільшенню продажів та дозволяє мінімізувати проблему відсутності на складі. До того ж, кожен артикул має бути в наявності і на своєму місці. Але не так просто визначити оптимальне розташування кожного продукту. Ідеальне розміщення знаходиться на перетині інтересів постачальника, ритейлера та покупців.

Розглянемо методи, які можуть бути використані для оптимального формування списку товарів.

Байєсівські мережі (БМ) – це графи із деякими характерними властивостями. Ідея впровадження БМ полягає у представленні причинно-наслідкових зв'язків, характерних для процесу у вигляді графа [13].

Тотожним найменуванням методу байєсівських мереж є поняття «байєсівська класифікація». Спочатку байєсівська класифікація використовувалася в експертних системах для формалізації знань. У сучасному світі байєсівська класифікація застосовується як один з методів Data Mining. Data Mining означає знаходження даних, їх глибинний аналіз. Data Mining використовується і для позначення системи методів знаходження статистичних даних практично корисних знань, невідомих раніше, необхідних для прийняття рішень у різних сферах діяльності людей [14].

Розрізняють такі типи байєсівських мереж [15]:

1) Дискретні БМ – мережі, у яких змінні вузлів представлені дискретними величинами. Дискретні БМ мають такі властивості:

- кожна вершина представляє собою подію, що описується випадковою величиною, яка може мати кілька станів;
- всі вершини, пов'язані з «батьківськими», визначаються таблицею умовних ймовірностей (ТУЙ) або функцією умовних ймовірностей;
- для вершин без «батьків» ймовірності їх станів є безумовними (маргінальними).

2) Динамічні БМ – мережі, у яких значення вузлів змінюються з часом, тобто це мережа, яка описує стани динамічної системи.

3) Неперервні БМ – мережі, в яких змінні вузлів – це неперервні величини. У багатьох випадках події можуть приймати будь-які стани з деякого діапазону. Тобто змінна X – неперервна випадкова величина, простором можливих станів якої є весь діапазон її допустимих значень $X = \{x | a \leq x \leq b\}$, що містить нескінченну множину точок.

4) Гібридні БМ – мережі, які містять вузли з дискретними і неперервними змінними. Але при використанні БМ, що містять неперервні і дискретні змінні, існує певний ряд обмежень: дискретні змінні не можуть мати неперервних батьків; неперервні змінні повинні мати нормальний закон розподілу, умовний на значеннях батьків; розподіл неперервної змінної X з дискретними батьками Y та неперервними батьками Z є нормальним.

Головною перевагою БМ виділяють здатність виявляти невідомі та нетривіальні зв'язки між факторами, про які експерти в предметній галузі не завжди можуть мати уявлення. Байєсівські мережі знаходять своє практичне застосування у таких сферах, як медицина, фінанси, економіка, комп'ютери і системне програмне забезпечення, обробка зображень та відео, космічні польоти та дослідження тощо. Особливістю БМ серед інших методів інтелектуального аналізу даних є те, що вони дозволяють враховувати та використовувати вхідні дані у вигляді експертних оцінок і статистичної інформації. Змінні можуть бути дискретними і неперервними, а характер їх надходження при аналізі та прийнятті рішення можливий в режимі реального часу та у вигляді статичних масивів

інформації і баз даних. Завдяки використанню представлення взаємодії між факторами процесу у вигляді причинно-наслідкових зв'язків у мережі досягається достатньо високий рівень візуалізації та чітке розуміння взаємодії факторів між собою. Ще одною перевагою БМ є можливість врахування невизначеностей різного характеру та формування висновку за допомогою наближених та точних методів. Але серед такої кількості переваг існують і недоліки, а саме:

- визначення всіх взаємодій у мережах Байєса для складних систем не завжди можна здійснити;
- підхід Байєса вимагає знання безлічі умовних ймовірностей, які зазвичай одержують експертними методами. Застосування програмного забезпечення ґрунтується на експертних оцінках.

Метод експертних оцінок застосовується при розробці середньострокових і довгострокових прогнозів, наприклад, при прогнозі структури споживчого попиту. Найбільш ефективним є застосування експертних методів в поєднанні з статистичними методами. Самостійне використання експертних оцінок рекомендується для випадків, коли повністю відсутня інформація, що дозволяє здійснювати кількісні розрахунки. Експерт – висококваліфікований фахівець, який знає особливості формування і розвитку об'єктів, який здатний дати досить надійні кількісні оцінки. Якість експерта – сукупність показників, які характеризують уміння експерта з достатньою точністю оцінювати розвиток об'єкта. Тур експертизи – частина процедури, яка включає підготовку опитування експертів, проведення опитування і обробку його результатів. Робоча група – група спеціалістів, які реалізують організацію проведення експертизи [16].

Найбільш застосовуваними процедурами експертних вимірів є: ранжування; парне порівнювання; множинні порівняння; безпосередня оцінка.

Рангом називається ступінь відмінності по якійсь ознаці, а ранжуванням – процес визначення рангів, відносних кількісних оцінок ступенів відмінностей за

якісними ознаками (наприклад, розташування факторів у порядку їх суттєвості, значимості в даному дослідницькому контексті). Ранжування застосовується у випадках, коли неможлива або недоцільна безпосередня оцінка. При цьому ранжування об'єктів містить лише інформацію про те, який з об'єктів кращий, і не містить інформацію про те, наскільки або у скільки разів один об'єкт переважає інший [17].

Метод полягає в тому, що експерти розташовують об'єкти ранжування (наприклад, критерії) в порядку зниження їх значущості (для альтернатив це спадання переваги).

Ранжування доцільно застосовувати, коли недоцільне або неможливе безпосереднє оцінювання. Метод передбачає присвоєння об'єкту значення за рейтинговою шкалою (присвоєння об'єкту оцінки за певний проміжок). Наприклад, оцінка від 0 до 100 на основі переваги щодо певних характеристик або їх груп.

Якщо є n об'єктів, то в результаті їх ранжування j -им експертом кожен об'єкт отримує оцінку x_{ij} – ранг, що приписується i -му об'єкту j -им експертом. Значення x_{ij} знаходяться в інтервалі від 1 до n . Ранг найважливішого показника дорівнює одиниці, найменш значущого – числу n . Ранжуванням j -го експерта називається послідовність рангів $x_{1j}, x_{2j}, \dots, x_{nj}$.

Застосування строгих чисельних відношень «більше» ($>$), «менше» ($<$) або «дорівнює» ($=$) не завжди дозволяє встановити порядок об'єктам. Поряд з ними використовуються відношення для визначення більшою чи меншою мірою будь-якої якісної ознаки (відношення часткового порядку, наприклад корисності), відношення типу «Більш переважно», «Менш переважно» «Рівноцінно» або «Байдуже».

Таке впорядкування утворює нестрогий лінійний порядок. Аналогічний вигляд має таблиця, якщо здійснюється ранжування об'єктів одним експертом за декількома показниками порівняння, в таблиці замість експертів в графах вказуються показники. Ранги об'єктів визначають тільки порядок розташування

об'єктів за показниками порівняння. Ранги як числа не дають можливості зробити висновок про те, наскільки або у скільки разів краще один об'єкт в порівнянні з іншим.

До недоліку цього методу можна віднести обмежені можливості використання, що обумовлюється складністю його використання при оцінюванні великої кількості об'єктів. Головною перевагою такого метода є простота та швидкість виконання. Оскільки поставлена задача вимагає швидкого та ефективного рішення буде доцільно використовувати метод ранжування, який допоможе найбільш коректно, ефективно та швидко сформувати список товарів для супермаркету.

2.2 Розробка математичної моделі формування списку товарів на сенсорні ваги самообслуговування

Імпульсні покупки складають найпривабливішу групу з точки зору можливостей прикладного застосування мерчандайзингу і VTL-технологій. Особливо важлива для незапланованих (імпульсних) покупок експозиція (виставка, викладка асортиментної позиції) в точці продажу. Відповідно до теорії психології 70% рішень людина приймає під впливом причин, які виключають участь логічного мислення. Здійснення імпульсної покупки – процес явного домінування бажання над розумом, коли рішення приймається миттєво, без осмислення сильних і слабких сторін товару. VTL-технології освоюють все більше продуктивних виробників, і їх число буде рости в міру зростання конкуренції. Зростанню VTL сприяє і підвищення рівня професіоналізму агентств, які пропонують свої послуги в цій сфері. Для ритейлу імпульсні покупки сьогодні мають важливе значення. Кількість товарів імпульсного попиту в мережах міні маркетів може становити до 50% обороту, в супер- і гіпермаркетах – до 20%. Тому торгові мережі мають бути зацікавлені у співпраці з виробниками товарів в проведенні заходів, пов'язаних з просуванням [18].

Для ефективної роботи супермаркету велику роль має порядок розташування товарів, акційні пропозиції, що привертають увагу і звичайно актуальна інформація про товар, оскільки кожен покупець буде не задоволений, коли на касі його сума чеку буде більша ніж він розраховував. В цьому розділі буде запропоновано алгоритм, що дозволяє значно знизити витрати часу на формування списку товарів та даних про них за рахунок методики підбору товарів.

Буде доцільним для поставленої задачі використовувати метод ранжування. Рангом називається ступінь відмінності за деякою ознакою, а ранжуванням – процес визначення рангів, відносних кількісних оцінок ступенів відмінностей за якісними ознаками (наприклад, розташування факторів у порядку їх суттєвості, значимості в даному дослідницькому контексті). Ранжування застосовується у випадках, коли неможлива або недоцільна безпосередня оцінка. При цьому «ранжування об'єктів» містить лише інформацію про те, який з об'єктів кращий, і не містить інформацію про те, наскільки або у скільки разів один об'єкт переважає інший.

Метод полягає в тому, що експерти розташовують об'єкти ранжування (наприклад, критерії) в порядку зниження їх значущості (для альтернатив це спадання переваги).

Метод полягає у віднесенні оціненого об'єкта до певного значення за оціночною шкалою (в присвоєнні об'єкту оцінки балів в певному інтервалі). Наприклад, оцінка від 0 до 10 згідно із перевагою за деякою ознакою або їх групі (наприклад, альтернативі – за перевагою, критерію – за значущістю, фактором зовнішнього середовища – за впливом, проблемі – за пріоритетністю рішення) [19].

Для підбору потрібних товарів, виокремимо такі характеристики:

1. Найменша кількість днів, що може зберігатись товар (для нормального обороту магазину, завжди першими потрібно продати товар, який може швидко зіпсуватись).

2. Товар, який найдовше не купували (потрібно знайти товар, який давно не купували та звернути на нього увагу).

3. Кількість днів, поки товар вважається сезонним.

Введемо такі позначення: I – кількість днів, найменшого терміну зберігання; J – найбільша кількість днів товару, що не продається; R – кількість днів, що товар вважається сезонним; A_i – кількість товарів, які швидко псуються $i, i \in I$; B_j – кількість товарів, які не продаються $j, j \in J$; $X_{ij}(N)$ – кількість днів, між моментами продажів N і $N+1$ (нумерація ведеться починаючи від останнього), для товару $i, i \in I, j \in J$. Додамо, що $X_{ij}(0)$ – кількість днів від останньої дати до дати останнього продажу; Z_{ij}^r – бінарна ознака, що сигналізує про те, що товару i фізично не було на залишках магазину j в день $r, i \in I, j \in J, r \in R$. Значення цього показника стає відомим тільки після проведення перевірки товару:

$$\Sigma_{ij}^r = \begin{cases} 1, \text{товар } i \text{ був відсутній на складі в момент } r, \\ 0, \text{товар } i \text{ був присутній на складі в момент } r. \end{cases} \quad (2.1)$$

де Σ_{ij}^r – кількість товару i , що є на складі в момент часу r .

$$C_{ij}^r = \frac{X_{kj}^r(0), k \in A_i}{\max_{k \in A_i} \{X_{kj}^r(0)\}}, \quad (2.2)$$

де C_{ij}^r – середня кількість днів з дня останньої продажі товару i в момент часу $r, i \in I, j \in J, r \in R$.

$$D_{ij}^r = \frac{X_{kj}^r(0), k \in A_i}{\max_{k \in A_i} \{X_{kj}^r(0)\}}, \quad (2.3)$$

де D_{ij}^r – середня кількість днів з дня останньої продажі товару i в магазинах, аналогічних j , в момент часу $r, i \in I, j \in J, r \in R$.

$$E_{ij}^r = \frac{X_{kj}^r(N), N=1, S}{\max_{N=1, S} \{X_{kj}^r(0)\}}, \quad (2.4)$$

де $E_{ij}^r(S)$ – середня кількість днів між продажами товару i в магазині j за останні S за останні S раз продаж, починаючи з моменту останньої продажі (число S задається експертним шляхом).

Таким чином, потрібно знайти деяку функцію, яка говорить про те, що потрібно продавати саме цей товар або ні:

$$F_{ij} = \begin{cases} 1, & \text{товар потрібно продати,} \\ 0, & \text{товар може зачекати на складі.} \end{cases} \quad (2.5)$$

Індекс r в функції не використовується, оскільки її результат актуальний тільки на поточний момент. Зауважимо, що для різних i, j значення показників можуть бути не визначені. Наприклад, якщо у товару відсутні аналоги, тоді D_{ij}^r не існує для будь-яких j і r .

Для знаходження кінцевої функції F_{ij} залишилось знайти оцінки коефіцієнтів α_1 , α_2 і α_3 які забезпечують максимальну точність прогнозування. Для цього, використовуючи історичні дані за значеннями показників і результати перерахунку вибраних товарів, вирішимо оптимізаційну модель:

$$\sum_{r \in R} \sum_{i \in I} \sum_{j \in J} (|L_{ij}^r - Z_{ij}^r| \cdot Z_{ij}^r) \rightarrow \min_{\alpha_1 \alpha_2 \alpha_3}. \quad (2.6)$$

Позначимо отримані оцінки як α'_1 , α'_2 і α'_3 . Тоді шукана функція виглядає так:

$$F_{ij} = \begin{cases} 1, & \alpha'_1 \cdot C_{ij}^t + \alpha'_2 \cdot C_{ij}^t + \alpha'_3 \cdot E_{ij}^t(S) \geq 1, \\ 0, & \alpha'_1 \cdot C_{ij}^t + \alpha'_2 \cdot C_{ij}^t + \alpha'_3 \cdot E_{ij}^t(S) < 1. \\ \dots \end{cases} \quad (2.7)$$

Отже, було розроблено математичну модель для інформаційної технології автоматизації роботи супермаркетів, яка підвищить ефективність формування списку товарів та прискорить швидкодію процесу обмінів між інформаційною базою та вагами самообслуговування.

2.3 Розробка UML-діаграми класів інформаційної технології автоматизації роботи супермаркетів

UML (англ. Unified Modeling Language) – уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, яка називається UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація [20].

Як будь-яка інша мова, UML має власні правила оформлення моделей і синтаксис. За допомогою графічної нотації UML можна візуалізувати систему, об'єднати всі компоненти в єдину структуру, уточнювати і покращувати модель у процесі роботи. На загальному рівні графічна нотація UML містить 4 основні типи елементів: фігури; лінії; значки; написи. UML-нотація є галузевим стандартом у сфері розробки програмного забезпечення, IT-інфраструктури і бізнес-систем [21].

Клас у мові UML слугує для позначення множини об'єктів, що мають однакову структуру, поведінку і відношення з об'єктами з інших класів. Графічно клас зображують у вигляді прямокутника, який додатково може бути розділений горизонтальними лініями на розділи або секції. У цих розділах можуть вказуватися ім'я класу, атрибути (дані) і операції (методи) [22].

Програмне забезпечення автоматизації роботи супермаркетів, містить такі основні класи: Settings, CategoriesGoods, Goods, User, Nomenclature, Libra, Price, BarCode та TypePrice.

Класи CategoriesGoods, Goods, User, Nomenclature, Libra, Price, BarCode та TypePrice слугують для маніпуляції даних, які отримані з БД. Всі ці класи наслідуються від класу Settings, який реалізує маркерний інтерфейс. Завдяки цьому класи можна реалізувати та зберігати поточний стан системи.

Клас User має набір публічних методів для роботи з БД. Метод find Client DB служить для авторизації та збереження налаштувань, цей метод приймає на вхід дві змінних літерного типу, ім'я користувача та права доступу. Метод повертає екземпляр класу User – авторизований користувач.

Клас Settings зберігає налаштування зроблені користувачем та має метод choice, що відкриває раніше збережені налаштування та метод save, що зберігає будь-які зміни в програмному додатку.

Клас Libra являє собою ваги на які будуть завантажуватись товари, метод Scale Availability перевіряє доступність вагів для завантаження, якщо ваги доступні метод Unload завантажує потрібну інформацію.

Клас CategoriesGoods – це категорії на які поділені всі товари та які будуть відображатись на вагах, метод Link Products дає змогу обрати товари в категорію, або видалити вже існуючі та повертає об'єкт класу Goods.

Метод Create product buttons відображає обрані нами товари на емуляторі екрану вагів.

Метод commit And Close викликається після редагування БД для застосування змін.

Клас Price, BarCode та TypePrice потрібні для завантаження коректної інформації та роздрукування етикетки для будь-якого товару на вагах.

На рис. 2.1 подана UML-діаграма асоціацій класів інформаційної технології автоматизації роботи супермаркетів.

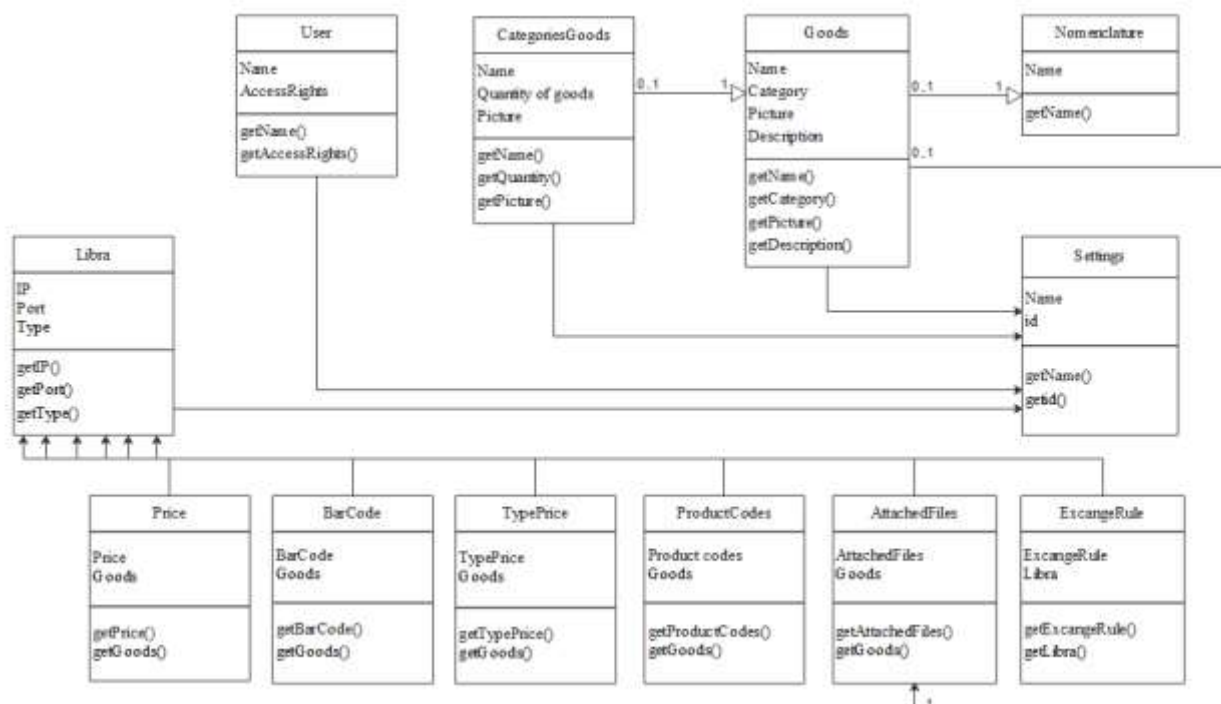


Рисунок 2.1 – UML-діаграма асоціацій класів інформаційної технології автоматизації роботи супермаркетів

Отже, було виділено основні класи інформаційної технології автоматизації роботи супермаркетів та змодельовано UML-діаграму асоціацій класів інформаційної технології автоматизації роботи супермаркетів.

2.4 Розробка ER-моделі бази даних автоматизації роботи супермаркетів та обґрунтування вибору бази даних

Для моделювання бази даних інформаційної технології для автоматизації роботи супермаркетів було обрано реляційну модель даних, оскільки у реляційній моделі досягається більш високий рівень абстракції даних, ніж в ієрархічній або мережевій.

Реляційна (від англ. relation) база даних – це тип бази даних, що зберігає інформацію в електронних таблицях і здійснює пошук даних в одній таблиці на підставі визначених ключових полів іншої таблиці [23].

Реляційна база даних – база даних, побудована на основі реляційної моделі. У реляційній базі кожен об'єкт задається відповідним записом (рядком) в таблиці. Реляційна база створюється та потім управляється за допомогою реляційної системи управління базами даних. Фактично реляційна база даних це інформація, що зберігається в двовимірних таблицях. Зв'язок між таблицями може знаходити своє відображення в структурі даних, а може тільки матися на увазі, тобто бути присутнім на не формалізованому рівні. Кожна таблиця БД представляється як сукупність рядків і стовпців, де рядки відповідають екземпляру об'єкта, конкретної події або явища, а стовпці – атрибутам (ознаками, характеристиками, параметрами) об'єкта, події, явища. Реляційні бази даних надають більш простий доступ до складання звітів (зазвичай через SQL) і забезпечують підвищену надійність і цілісність даних завдяки відсутності надлишкової інформації [24].

Зв'язок встановлюється між двома загальними полями (стовпцями) двох таблиць. Існують зв'язки з відношенням «один-до-одного», «один-до-багатьох» і «багато-до-багатьох».

Відносини, які можуть існувати між записами двох таблиць [24]:

- один-до-одного – кожному запису з однієї таблиці відповідає один запис у іншій таблиці;
- один-до-багатьох – кожному запису з однієї таблиці відповідає декілька записів у іншій таблиці;
- багато-до-одного – безлічі записів з однієї таблиці відповідає один запис у іншій таблиці;
- багато-до-багатьох – безлічі записів з однієї таблиці відповідає декілька записів в іншій таблиці.

Відношення «один-до-багатьох» створюється в тому випадку, коли тільки одне з полів є полем первинного ключа або унікального індексу.

Відношення «один-до-одного» створюється в тому випадку, коли обидва поля є ключовими або мають унікальні індекси.

Відношення «багато-до-багатьох» фактично є двома відносинами «один-до-багатьох» з третьої таблицею, первинний ключ якої складається з полів зовнішнього ключа двох інших таблиць.

Обрано цю модель, тому що вона має такі переваги:

- доступність для розуміння та простота. Інформаційною конструкцією є таблиця;
- суворі правила проектування, що базуються на математичному апараті;
- незалежність даних, тобто при зміні БД зміни в програмі мінімальні;
- для організації запитів та написання програмного забезпечення не потрібно знати конкретну організацію БД у зовнішній пам'яті.

Перед тим, як почати розробляти базу даних для інформаційної технології автоматизації роботи супермаркетів, потрібно визначити сутності, які будуть зберігатись та накопичуватись в базі даних. Перш за все потрібно зберігати дані користувачів, які мають працювати з технологією (для авторизації та обмеження прав в подальшій роботі). Потрібно мати дані організації (супермаркету) з яким ми працюємо. Тоді нам потрібно мати інформацію про товари (номенклатура), кількість залишків, ціну, знижку, сезонність, термін зберігання, картинки, штрих-коди. Ваги з якими програмне забезпечення робить обміни (IP, порт, тип вагів). Налаштування обмінів. Звіти з продажів.

Такі сутності підібрані так, щоб забезпечити повну функціональність роботи інформаційної технології автоматизації роботи супермаркетів.

Отже, сутності в базі даних програмного продукту для інформаційної технології автоматизації роботи супермаркетів: користувачі, організація, товари, ціни, ваги, налаштування обмінів.

Атрибути сутностей:

1) Користувачі (ПІБ користувача, організація, адреса користувача, номер телефону, електронна адреса користувача, посада користувача, логін користувача, пароль користувача).

2) Організація (назва організації, юридична адреса організації, ЄДРПОУ організації).

3) Номенклатура (назва номенклатури, кількість залишків номенклатури, одиниця виміру номенклатури, сезонність номенклатури, термін зберігання номенклатури, штрих-код номенклатури, картинка номенклатури, ціна номенклатури).

4) Ціни (сума, тип ціни, номенклатура).

5) Ваги (назва вагів, IP вагів, порт вагів, тип вагів).

6) Налаштування обмінів (назва налаштувань, дата створення, користувач, організація, товари, ціни, ваги, шлях для збереження).

Вкажемо універсальне відношення для бази даних інформаційної технології автоматизації роботи супермаркетів: «R»: (ПІБ користувача, адреса користувача, номер телефону, електронна адреса користувача, посада користувача, логін користувача, пароль користувача, назва організації, юридична адреса організації, ЄДРПОУ організації, назва номенклатури, кількість залишків номенклатури, одиниця вимірювання номенклатури, сезонність номенклатури, термін зберігання номенклатури, штрих-код номенклатури, картинка номенклатури, ціна номенклатури, сума, тип ціни номенклатури, назва вагів, IP вагів, порт вагів, тип вагів, назва налаштувань, дата створення, шлях для збереження). Ступінь універсального відношення – 27.

Характеристики зв'язків бази даних інформаційної технології автоматизації роботи супермаркетів представлена в табл. 2.1.

У звичайних випадках для побудови концептуальної схеми використовуються звичайні методи агрегації та узагальнення. При агрегації декілька інформаційних об'єктів (елементів даних) об'єднується в один у відповідності з семантичними зв'язками між об'єктами. При побудові інфологічних моделей можна використовувати мову ER-діаграм (англ. EntityRelationship, "сутність-зв'язок"). Тоді сутності зображуються позначеними прямокутниками, асоціації - позначеними ромбами або шестикутниками,

атрибути - позначеними овалами, а зв'язки між ними позначаються ненаправленими ребрами, над якими може проставлятися ступінь зв'язку (1 або «Б») і необхідне пояснення. Зв'язки зображують у вигляді ліній: прямою - обов'язковий зв'язок, пунктирною - необов'язковий зв'язок [25].

Таблиця 2.1 – Характеристики зв'язків бази даних інформаційної технології автоматизації роботи супермаркетів

Ім'я сутності 1	Ім'я зв'язку	Ім'я сутності 2	Тип зв'язку
Організація	Має	Користувачів	Б:Б
Організація	Має	Ваги	Б:Б
Організація	Має	Налаштування обмінів	1:Б
Користувачі	Створюють	Номенклатуру	Б:Б
Користувачі	Вказують	Ціну	Б:Б
Номенклатура	Має	Ціну	1:1
Користувачі	Створюють	Налаштування обмінів	Б:1
Номенклатура	Входить до	Налаштування обмінів	Б:1
Ціна	Входить до	Налаштування обмінів	Б:1
Ваги	Входить до	Налаштування обмінів	Б:1

Представимо ER-модель для бази даних інформаційної технології автоматизації роботи супермаркетів на рис. 2.2.

Дані, які визначають логіку функціонування системи на базі 1С: Підприємства, відносяться до інформаційної бази. Зберігання інформаційної бази здійснюється в базі даних у вигляді набору таблиць, для чого 1С: Підприємство 8 може використовувати одну з п'яти систем управління базами даних (СУБД) [26]:

– вбудовану в 1С: Підприємство 8 (файловий варіант інформаційної бази). В цьому випадку всі дані інформаційної бази зберігаються в файлі з ім'ям

1cv8.1cd. Цей файл має двійковий формат і по суті є базою даних для вбудованої в 1С: Підприємство 8 СУБД;

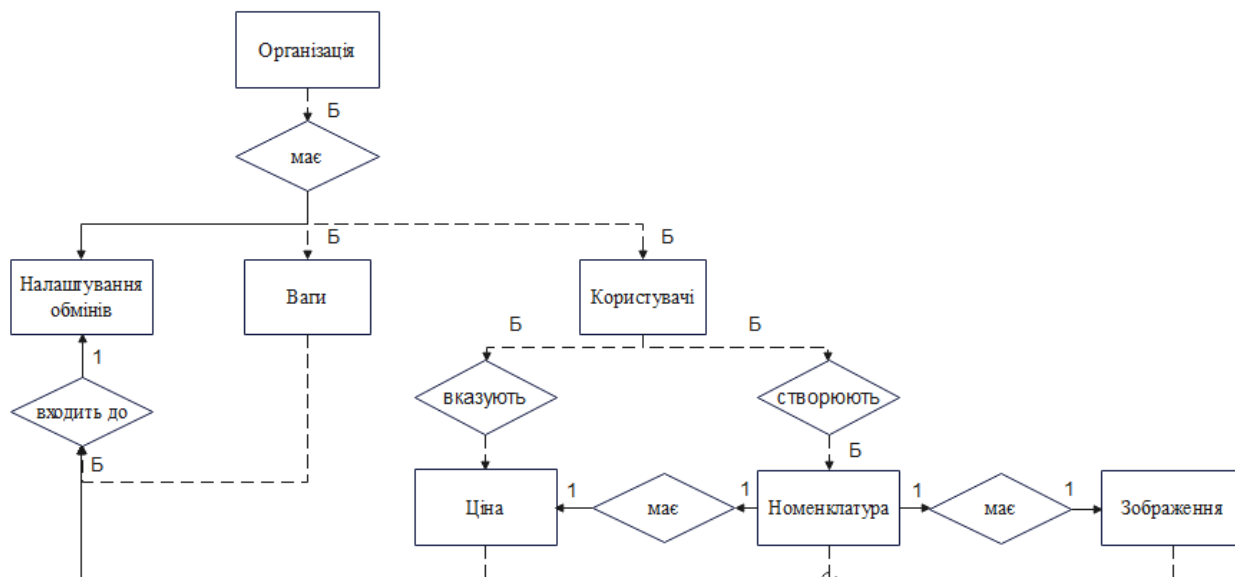


Рисунок 2.2 – Схеми ER-моделі бази даних інформаційної технології автоматизації роботи супермаркетів

- Microsoft SQL Server (клієнт-серверний варіант інформаційної бази). Всі дані інформаційної бази зберігаються в базі даних Microsoft SQL Server;
- PostgreSQL (клієнт-серверний варіант інформаційної бази). Всі дані інформаційної бази зберігаються в базі даних PostgreSQL;
- IBM DB2 (клієнт-серверний варіант інформаційної бази). Всі дані інформаційної бази зберігаються в базі даних IBM DB2;
- Oracle Database (клієнт-серверний варіант інформаційної бази). Всі дані інформаційної бази зберігаються в базі даних Oracle Database;

Інформаційна база подана в базі даних у вигляді таблиць. Серед таких таблиць є обов'язкові, а саме:

- Config – основна конфігурація інформаційної бази. Ця конфігурація відповідає реальній структурі даних та використовується 1С:Підприємством 8.0 в режимі Підприємства;

– ConfigSave – конфігурація, редагована конфігуратором. Конфігурація з ConfigSave переписується в Config під час виконання "Оновлення конфігурації бази даних" у Конфігураторі, а навпаки - під час виконання в Конфігураторі операції "Конфігурація - Конфігурація бази даних - Повернутися до конфігурації бази даних";

– Files – містить службову інформацію, наприклад, про роботу зі сховищем конфігурації;

– Params містить в собі параметри інформаційної бази, а саме:

- національні налаштування (мова);
- список користувачів інформаційної бази;
- відповідності між метаданими та об'єктами бази даних;
- інші налаштування.

– UsersWorkHistory містить історію роботи користувачів програми;

– _YearOffset – зміщення дат у базі даних. Ця таблиця створюється лише під час використання Microsoft SQL Server;

– DBSchema містить інформацію про структуру бази даних 1С:Підприємства та визначає інші об'єкти бази даних, що використовуються даною інформаційною базою;

- v8users – користувачі програми;
- SystemSettings – це сховище системних налаштувань;
- RepVarSettings – налаштування варіантів звітів;
- FrmDtSettings – налаштування даних форм;

Об'єкти метаданих, яким можуть відповідати ті чи інші таблиці.

Константи: _Consts містить поточні значення всіх констант, визначених конфігурації.

Плани обміну: _Node<n> – таблиця плану обміну.

Довідники: _Reference<n> – таблиця довідника.

Документи: `_Document<n>` – таблиця документів кожного об'єкта метаданих " документ " .

Послідовності документів: `_Sequence<n>` – таблиця реєстрації документів - кожної послідовності.

Журнали документів: `_DocumentJournal<n>` – таблиця журналу документів, що створюється для кожного журналу документів.

Перерахування: `_Enum<n>` – таблиця перерахування - за однією для кожного перерахування.

Плани видів характеристик: `_Chrc<n>` – основна таблиця плану видів показників.

Реєстри відомостей: `_InfoReg<n>` – таблиця рухів реєстру відомостей.

Реєстри накопичення: `_AccumReg<n>` – таблиця рухів реєстру накопичення.

Реєстри бухгалтерії: `_AcctReg<n>` – таблиця рухів реєстру бухгалтерії.

Реєстри розрахунку: `_CalcReg<n>` – таблиця рухів реєстру розрахунку.

Бізнес процеси: `_BPRoutePoint<n>` – таблиця точок маршруту бізнес-процесу для кожного бізнес-процесу.

Завдання: `_Task<n>` – основна таблиця завдання.

База даних «1С: Підприємство» має деякі особливості в порівнянні з іншими класичними моделями управління базами даних. Головною відмінністю є те, що розробник не має безпосередньо доступу до бази даних, він працює виключно з платформою 1С: Підприємство 8. Це дає йому змогу описувати структури даних в конфігураторі, обробляти дані за допомогою вбудованих мовних об'єктів, робити запити до даних за допомогою мови запитів. Платформа 1С: Підприємство 8 забезпечує операції з виконання запитів, опису структур даних та обробки даних, перекладаючи їх у відповідні команди. Це можуть бути команди системи управління базою даних, якщо мова йде про версію роботи

клієнт-сервер, або команди власного механізму баз даних для версії файлу [27].
Загальна схема роботи платформи «1С: Підприємство» наведена на рис. 2.3.



Рисунок 2.3 – Загальна схема роботи платформи «1С: Підприємство»

База даних будь-якого бізнес-додатку має в собі величезну кількість інформації. Коли йдеться мова про програми для ведення бізнесу, в першу чергу мова йде про бухгалтерію, а така база даних має містити в собі: номенклатуру, контрагентів, документи, звіти, ціни, працівників, статистику тощо.

Особливістю бази даних в 1С є структуризація даних. Для того, щоб не накопичувати не потрібні дані, створено довідники, реєстри відомостей, реєстри накопичення, реєстри бухгалтерії, реєстри розрахунку.

Документи – це прикладні об'єкти конфігурації. Вони дозволяють зберігати в прикладному рішенні інформацію про скоєні господарські операції або події, що відбулися в житті підприємства взагалі. Це може бути, наприклад, прибуткові накладні, накази прийому працювати, рахунки, платіжні доручення тощо [28].

Довідники – це прикладні об'єкти конфігурації. Вони дозволяють зберігати в інформаційній базі дані, що мають однакову структуру та обліковий характер. Це може бути, наприклад, список співробітників, список товарів, список постачальників чи покупців. Застосування довідників у системах значно

прискорює процес занесення нової інформації. При цьому використання словника зменшує імовірність появи помилки під час заповнення полів форм, що система виводить на екран дисплея в процесі роботи. Крім того, указавши визначену позицію словника, можна зробити доступними системам усі характеристики цієї позиції при наступній обробці об'єкта обліку (документа, господарської операції і т.д.) [28].

Реєстри відомостей – це прикладні об'єкти конфігурації. Вони дозволяють зберігати у прикладному рішенні довільні дані у декількох вимірюваннях. Наприклад, у реєстрі відомостей можна зберігати курси валют у розрізі валют, або ціни підприємства у розрізі номенклатури та типу цін [29].

Реєстри накопичення – це прикладні об'єкти конфігурації. Вони становлять основу механізму обліку руху коштів (фінансів, товарів, матеріалів тощо), що дозволяє автоматизувати такі напрями, як складський облік, взаєморозрахунки, планування [30].

Реєстри бухгалтерії – це прикладні об'єкти конфігурації. Вони використовуються в механізмі бухгалтерського обліку та дозволяють вести багаторівневий та багатовимірний аналітичний облік, облік за декількома планами рахунків, опціональне ведення кількісного, сумового та валютного обліку за окремими розрізами аналітики тощо [31].

Реєстри розрахунку – це прикладні об'єкти конфігурації. Вони використовуються в механізмі складних періодичних розрахунків і служать для зберігання записів про ті чи інші види розрахунку, які необхідно виконати, а також для зберігання проміжних даних та результатів виконаних розрахунків [32].

Враховуючи ці особливості структури можемо визначити таке:

- Довідники: номенклатура, види цін, правила обміну з підключеним обладнанням, одиниці вимірювання, користувачі, організації, ваги, тип вагів;
- Реєстри накопичення: залишки номенклатури;
- Реєстри відомостей: картинки, штрих-коди, ціни;

На рис. 2.4 представлено зв'язки між таблицями бази даних інформаційної технології автоматизації роботи супермаркетів.

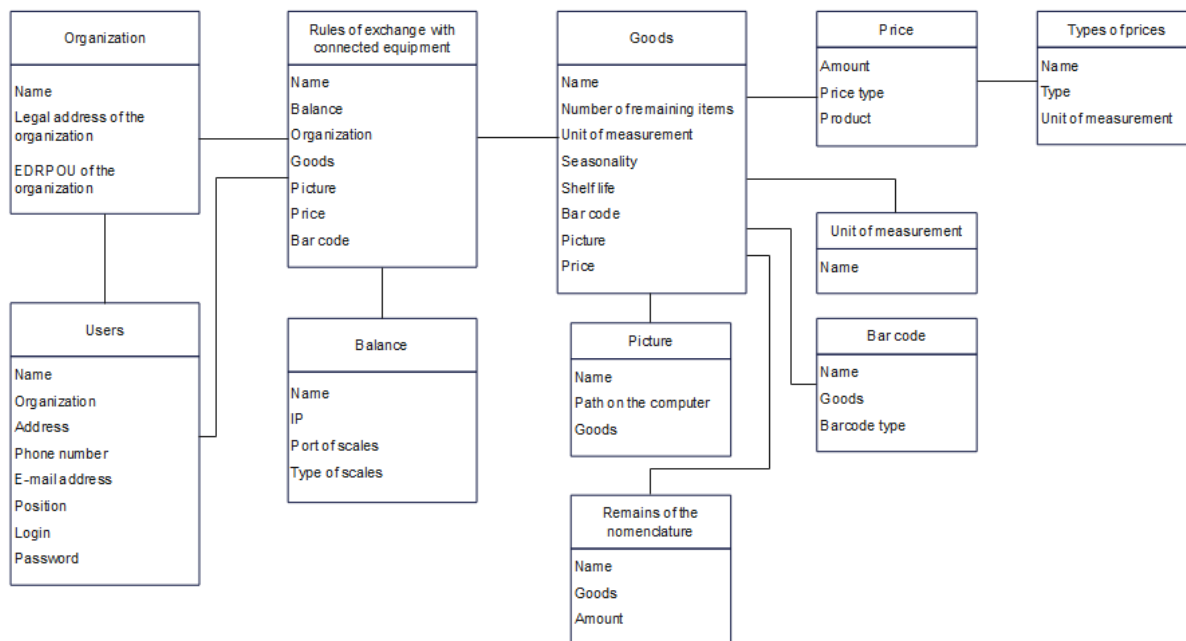


Рисунок 2.4 – Схема зв'язків у базі даних інформаційної технології автоматизації роботи супермаркетів

Враховуючи особливості реляційної моделі даних та структури платформи «1С: Підприємство», було виділено основні сутності, продемонстровано ER-модель та змодельовано схему зв'язків у базі даних інформаційної технології автоматизації роботи супермаркетів.

2.5 Висновок до розділу 2

У цьому розділі проаналізовано можливі методи формування списку товарів, а саме: Баєсові мережі та метод ранжування. Для подальшої роботи обрано метод ранжування, оскільки для такого метода характерні простота в реалізації, висока швидкодія, що і обумовлює його доцільність застосування в бізнес-сфері. Для розробки математичної моделі виокремлено характеристики:

найменша кількість днів, що може зберігатись товар; товар, який найдовше не купували; кількість днів, поки товар вважається сезонним, на основі який відбувалось подальше моделювання.

Продемонстровано UML-діаграму класів, виокремлено класи CategoriesGoods, Goods, User, Nomenclature, Libra, Price, BarCode та TypePrice, що слугують для маніпуляції даних, які отримані з БД.

Було обрано реляційну модель бази даних, оскільки вона проста для розуміння та має суворі правила проектування, які не суперечать структурі платформи «1С: Підприємство». Визначено сутності бази даних та їх зв'язки, а також спроектовано ER-модель бази даних та представлено зв'язки між таблицями бази даних інформаційної технології автоматизації роботи супермаркетів.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АВТОМАТИЗАЦІЇ РОБОТИ СУПЕРМАРКЕТІВ

3.1 Структурна організація інформаційної технології автоматизації роботи супермаркетів

BAS це логічний розвиток ідей та функціоналу, закладеного в програми 1С:Підприємство [33]. Тому для того, щоб розробити архітектуру BAS, потрібно з'ясувати, що собою являє 1С: Підприємство.

Структура програми «1С: Підприємство» включає такі поняття:

Платформа – це та база, яка складає основу усієї системи. Саме вона отримала назву «1С:Підприємство» та служить середовищем розробки прикладних рішень. З нею працюють програмісти [34].

Конфігурація – це безпосередньо та програма, яку розробляють фахівці на базі платформи «1С:Підприємство». Головна мета – виконання конкретних завдань обліку певного сектора на підприємстві, установі або організації. З часом кожна конфігурація удосконалюється, доповнюється та оновлюється. Прикладні рішення розвиваються, змінюються відповідно до нових вимог законодавства та вимог ринку [35].

База даних – це інформація, яку користувачі вводять до системи за допомогою інструментів, які надає певна конфігурація. Також, із допомогою тих самих інструментів, інформацію у базі даних можна обробити, відкоригувати, систематизувати, видалити тощо [35].

Найбільш концептуальними в архітектурі «1С: Підприємства» є сама наявність платформи і поняття бізнес-додатки. В «1С: Підприємство» було введено чіткий поділ на платформу і бізнес-додаток. Платформа являє собою так званий *framework*, в якому функціонує бізнес-додаток. Суттєвою перевагою чіткого розмежування між платформою і бізнес-додатком є високий рівень

адаптованості рішень під вимоги клієнта [36]. В основі бізнес-додатків лежать метадані. Метадані являють собою структурований декларативний опис бізнес-додатку. Вони формують ієрархію всіх об'єктів з яких формуються складові прикладної системи та визначають особливості поведінки.

У моделі розробки «1С: Підприємства» використовується підхід, опису метаданих у вигляді сукупності прикладних об'єктів, які обирають з певного набору прототипів (класів). Прикладами таких класів в «1С: Підприємство» є «Довідники», «Документи», «Перелічення», «Звіти», «Регістри накопичення», «Регістри відомостей», «Регістри бухгалтерії», «Регістри бухгалтерії», «Регістри розрахунку», «Бізнес процеси», тощо. Кожен такий прототип має базову реалізацію, яка визначає особливості функціонування, створюваних на основі такого прототипу об'єктів: структуру збережених елементів разом, набір типів мови програмування, методи, властивості і події, а також типові для розв'язуваної задачі операції, способи відображення і редагування, методи регулювання прав доступу тощо. Якщо в ході розробки необхідно створити об'єкт, що відображає особливості предметної галузі, програміст вибирає відповідний прототип і створює на його основі об'єкт метаданих [36].

Програмний додаток в 1С – це прототип, який входить в метадані та називається обробкою. Його переваги в тому, що він може бути вбудований в метадані платформи, а може бути зовнішнім та запускатись, як зовнішній файл. Цей програмний додаток користувач зможе використовувати так як йому буде зручно, а налаштування зроблені ним на своєму ПК, можна буде надіслати на будь-який інший та відкрити в такій же BAS Управління торгівлею (8.3.16.1690).

Для вдалої реалізації інформаційної технології автоматизації роботи супермаркетів, необхідно розробити структуру (рис. 3.1). Вона має бути зручною та інтуїтивно зрозумілою для працівників супермаркету, які будуть з нею працювати. Виокремлено такі модулі: модуль авторизації, модуль загальних налаштувань, модуль роботи з товарами, модуль роботи з вагами, модуль роботи з емулятор сенсорних вагів та модуль роботи з базою даних.

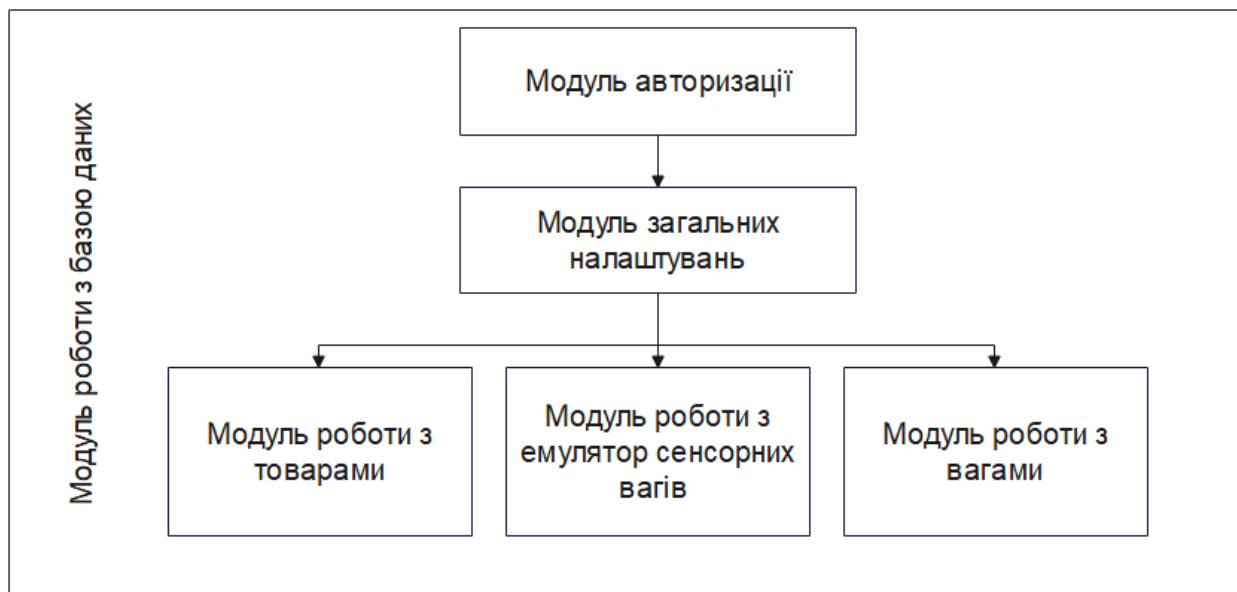


Рисунок 3.1 – Структурна схема інформаційної технології автоматизації роботи супермаркетів

В цьому програмному додатку будуть представлені модулі загальних налаштувань, модуль роботи з товарами, модуль додавання вагів та модуль роботи з базою даних.

Модуль загальних налаштувань є основним та містить у собі дані всіх інших модулів. В ньому користувач зможе зробити всі необхідні налаштування та контролювати процес завантаження товарів на ваги.

Модуль роботи з товарами надає змогу поділити товари на категорії, змінити зображення та обрати послідовність завантаження товарів на ваги.

Модуль додавання вагів потрібен для внесення всіх потрібних даних для підключення до вагів та роботи з ними. В ньому користувачеві потрібно вказати IP, порт та назву вагів.

Усі перераховані модулі пов'язані з модулем роботи з базою даних. Такий модуль забезпечує зв'язок з базою даних супермаркету, в якій є товари, штрихкоди, ціни та вся необхідна інформація для завантаження товарів на ваги.

3.2 Розробка схеми алгоритму роботи інформаційної технології автоматизації роботи супермаркетів

Алгоритм – це скінчена послідовність команд (вказівок), що визначає, які дії та у якому порядку потрібно виконати, щоб досягти поставленої мети [37]. Для розробки будь-якого програмного забезпечення потрібно мати набір послідовних інструкцій, що описують порядок дій та поведінку програми для досягнення цілі.

Розглянемо алгоритм роботи програмного забезпечення, який реалізує інформаційну технологію автоматизації роботи супермаркетів (рис. 3.2).

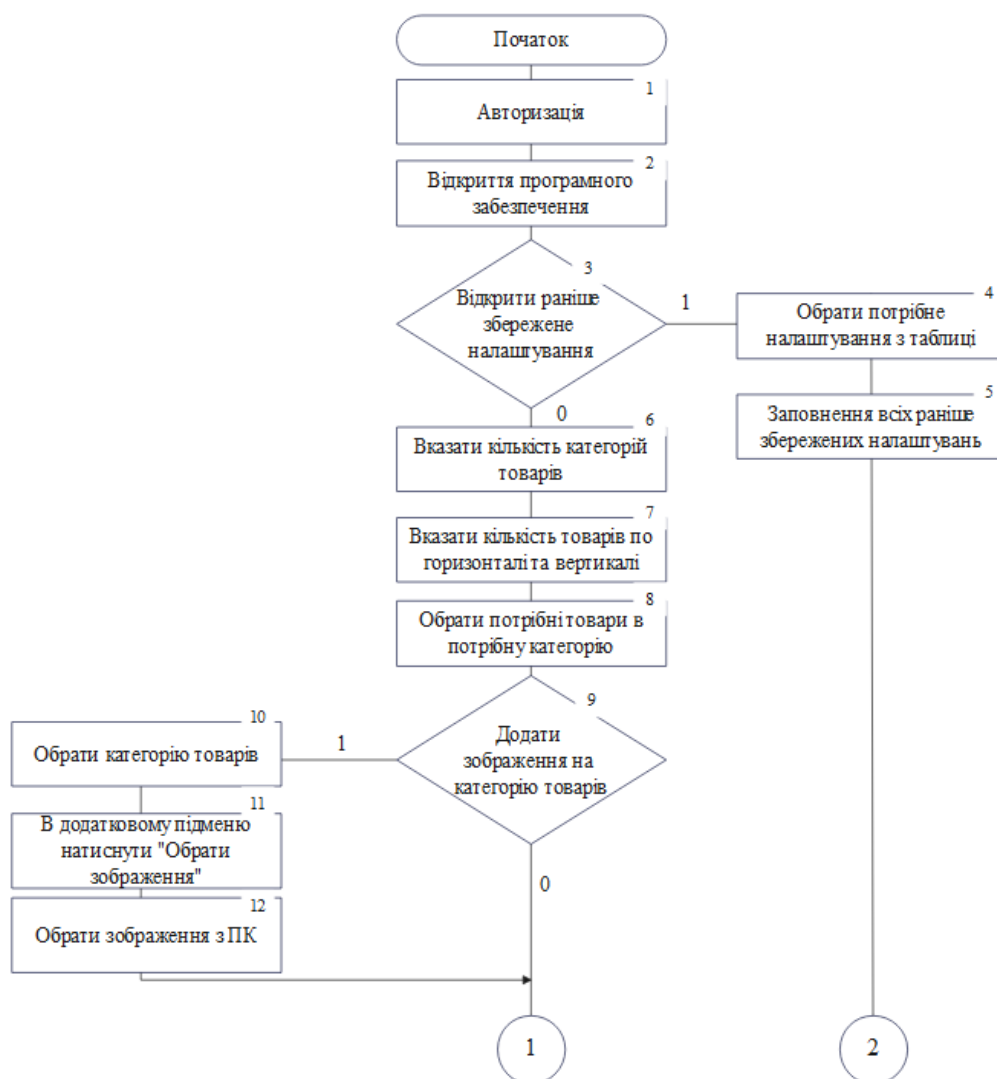


Рисунок 3.2 – Схема загального алгоритму функціонування інформаційної технології

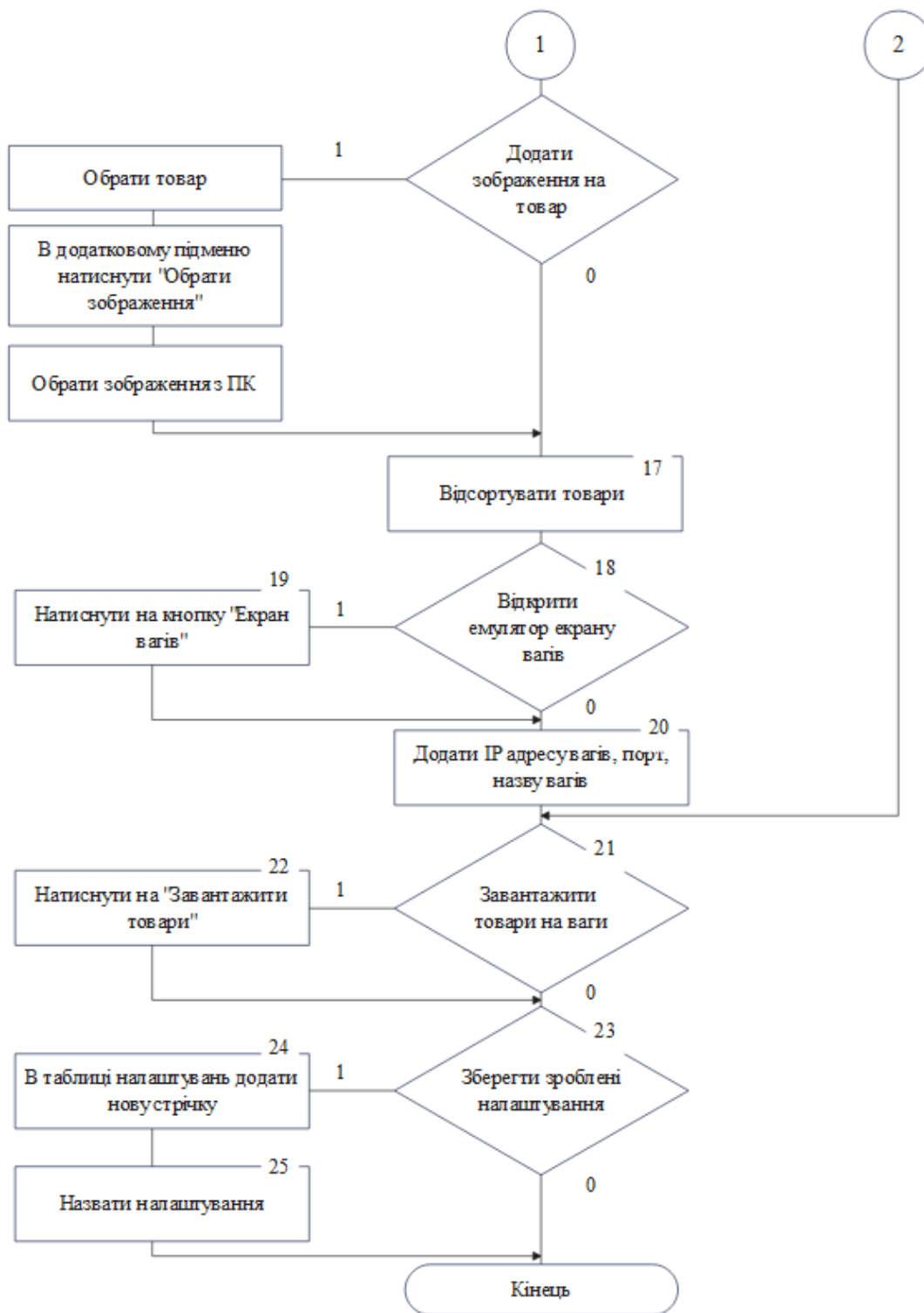


Рисунок 3.2, аркуш 2

Опишемо наведений алгоритм функціонування інформаційної технології, який подано на рис. 3.2:

Крок 1. На першому етапі користувач має авторизуватись для входу в BAS.

Крок 2. Відкрити програмне забезпечення для завантаження товарів на ваги.

Крок 3. В програмному забезпеченні представлена таблиця налаштувань, якщо користувач увійшов вперше вона пуста, якщо він раніше працював у програмному забезпеченні та зберіг свої налаштування він має можливість їх обрати. Якщо користувач їх обрав він переходить до Кроку 17, інакше переходить до кроку 4.

Крок 4. Для того щоб створити категорії товарів, йому необхідно вказати кількість необхідних категорій товарів.

Крок 5. Потрібно вказати розмір матричної сітки розміщення товарів на вагах, а саме кість товарів по горизонталі та по вертикалі.

Крок 6. Користувач має можливість змінити створені ним категорії товарів, а саме перейменувати та замінити картинку. Якщо користувач хоче замінити картинку категорії, він переходить до кроку 7, інакше до кроку 10.

Крок 7. Користувач бажає замінити зображення категорії товарів. Для цього необхідно обрати потрібну категорію.

Крок 8. Натиснувши на категорію, з'являється додаткове підменю з написом «Обрати зображення», яке потрібно натиснути.

Крок 9. Користувач обирає потрібне зображення на комп'ютері.

Крок 10. Користувач має можливість змінити стандартне зображення товару. Якщо користувач хоче замінити зображення товару, він переходить до кроку 11, інакше до кроку 14.

Крок 11. Користувач бажає замінити зображення товару. Для цього необхідно обрати необхідний товар.

Крок 12. Натиснувши на товар, з'являється додаткове підменю з написом «Обрати зображення», яке потрібно натиснути.

Крок 13. Користувач обирає потрібне зображення на комп'ютері.

Крок 14. Користувач має можливість перегляну вигляд товарів на вагах в програмному додатку. Якщо користувач хоче відкрити емулятор екрану вагів, він переходить до кроку 15, інакше до кроку 16.

Крок 15. Для відкриття емулятора екрану вагів потрібно натиснути «Екран вагів».

Крок 16. Потрібно вказати IP адрес, порт та назву вагів.

Крок 17. Якщо користувач хоче завантажити товари на ваги він переходить до кроку 18, інакше крок 19.

Крок 18. Натиснути «Завантажити товари».

Крок 19. Представлена можливість зберегти зроблені налаштування. Якщо користувач хоче зберегти налаштування він переходить до кроку 20, інакше крок 22.

Крок 20. В таблиці налаштувань додати новий рядок.

Крок 21. Назвати створене налаштування та зберегти.

Крок 22. Вихід з програмного додатку.

Схема алгоритму функціонування модуля сортування товарів перед завантаженням на сенсорні ваги самообслуговування інформаційної технології автоматизації роботи супермаркетів подано на рис. 3.3.

Опишемо наведений алгоритм функціонування модуля сортування товарів перед завантаженням на сенсорні ваги самообслуговування, який подано на рис. 3.3:

Крок 1. На першому кроці користувачеві представлена можливість відсортувати товари за алфавітом. Якщо користувач хоче відсортувати товари за алфавітом він переходить до кроку 2, інакше переходить до кроку 3.

Крок 2. Сортування товарів за алфавітом.

Крок 3. Користувачеві представлена можливість відсортувати товари за потребою в продажі. Якщо користувач хоче відсортувати товари за алфавітом він переходить до Кроку 4, інакше переходить до кроку 13.

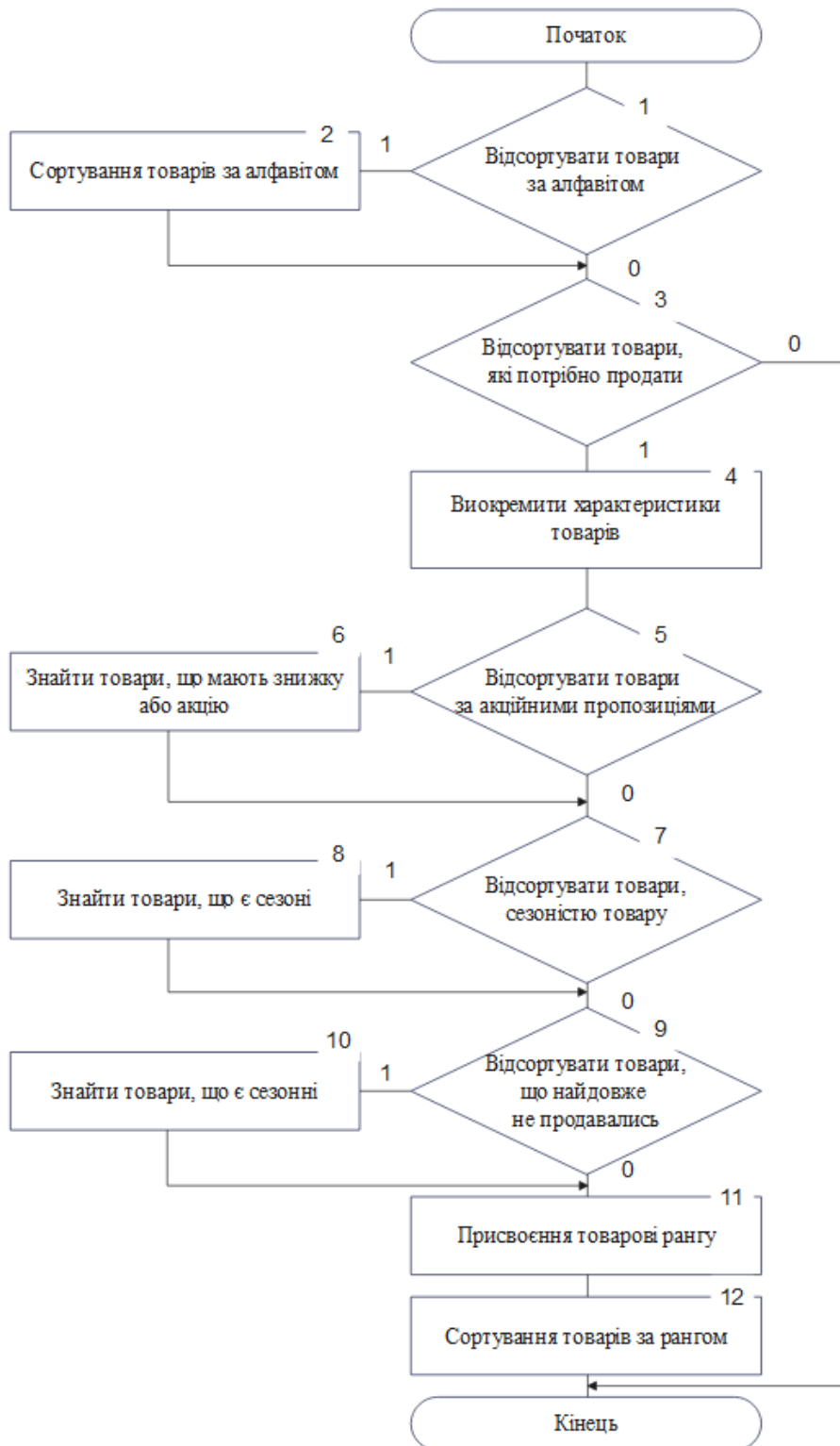


Рисунок 3.3 – Схема алгоритму функціонування модуля сортування товарів перед завантаженням на сенсорні ваги самообслуговування

Крок 4. Виокремлюються характеристики товарів, які слугують для сортування.

Крок 5. Представлена можливість відсортувати за акційними пропозиціями. Якщо користувач хоче відсортувати товари за таким критерієм він переходить до кроку 6, інакше до кроку 11.

Крок 6. Знаходження товарів із акційними пропозиціями чи знижкою.

Крок 7. Представлена можливість відсортувати за сезонністю товарів. Якщо користувач хоче відсортувати товари за цим критерієм він переходить до кроку 8, інакше до кроку 11.

Крок 8. Знаходження товарів, які є сезонними.

Крок 9. Представлена можливість відсортувати за кількістю днів, що не продавався товар. Якщо користувач хоче відсортувати товари за таким критерієм він переходить до кроку 10, інакше до кроку 11.

Крок 10. Знаходження кількості днів, що товар не продавався.

Крок 11. Присвоєння товарів рангу.

Крок 12. Сортування товарів за рангами.

Крок 13. Завершення сортування.

Отже, було розроблено схему алгоритма загального функціонування інформаційної технології та алгоритм функціонування модуля сортування товарів перед завантаженням на сенсорні ваги самообслуговування та описано кроки даних алгоритмів.

3.3 Обґрунтування вибору мови та середовища програмування

Для реалізації інформаційної технології доцільно використати об'єктно-орієнтовану мову програмування, порівняємо три мови програмування, що можуть вирішити поставлену задачу – C++, Java та 1C.

C++ – це універсальна мова програмування високого рівня з підтримкою декількох парадигм програмування. Зокрема: об'єктно-орієнтованої та

процедурної. При створенні C++ прагнули зберегти сумісність з мовою C. Більшість програм на C добре працюватимуть і з компілятором C++. C++ має синтаксис, заснований на синтаксисі C. Така мова з 1990-х стала однією із найпопулярніших мов програмування. Цю мову застосовують для розробки програмного забезпечення, написання драйверів, розробки серверних та клієнтських програм, написання ігор, тощо. Програми написані на C++ – мобільні, тому не прив'язані до комп'ютера певної марки, чи ОС [38]. До переваг можна віднести: висока швидкодія; гарна захищеність; може бути використана для розробки майже будь-якого продукту (web, desktop, etc.).

Недоліки: складна у написанні; високий поріг входження; низькорівнева; багатослівна у порівнянні з Python (через низькорівневність); невелика кількість бібліотек для штучного інтелекту та машинного навчання.

Java – популярна мова програмування, яка з'явилася в 1995 році й наразі налічує дев'ять основних версій. Java стала технічним феноменом, що багато в чому пов'язано з її унікальною портативністю: програми Java працюють на будь-якому пристрої або операційній системі [39]. Вперше була випущена Sun Microsystems в 1995 році. Досить суттєвою перевагою Java є велика колекція бібліотек з відкритим кодом та будь-який розробник може використовувати вже розроблений код. До переваг цієї мови програмування можна віднести [40]:

- легкість вивчення: цю мову можна швидко вивчити і домогтися високої продуктивності програмування. PHP призначений для Web-розробників і HTML-кодувальників, дозволяє їм без проблем додавати до своїх Web-сайтів сучасні можливості, такі як динамічна генерація сторінок;

- відкриті джерела: PHP розповсюджується за ліцензією Apache, яка передбачає комерційне і некомерційне використання і розробку. Це означає, що програмою можна вільно користуватися без відрахування ліцензійних зборів. Крім того, існує всесвітня мережа талановитих розробників, які постійно покращують і розвивають PHP. Завдяки доступності вихідного програмного коду можна налагодити програму або налаштувати її під свої потреби;

- підтримка баз даних: PHP забезпечує ефективну підтримку баз даних. Може працювати з ODBC, відкритими базами даних (MySQL і PostgreSQL), а також з комерційними (Microsoft SQL Server, Oracle і Sybase);
- розширення: існує безліч вільно доступних розширень і вихідного коду для будь-яких прикладних задач: від маніпуляції з XML до доступу до каталогів. Програмісти можуть використовувати цей готовий код для швидкого компілювання найсучасніших додатків;

Недоліки мови програмування:

- мова програмування в повному обсязі об'єктно-орієнтована: мова не має таких властивостей повністю об'єктно-орієнтованої мови, як індивідуальні змінні, множинне спадкування тощо;
- відповідність корпоративним вимогам: мова досить популярна у світі програм з відкритим кодом і технічно перевершує багато комерційних аналогів. Однак, якщо ми захочемо використовувати PHP в корпорації, то це або взагалі не вдасться зробити, або буде потрібно значно більше додаткових програмних засобів.

1С – мова програмування, яка використовується в сімействі програм «1С: Підприємство». Така мова є мовою, що інтерпретується мовою високого рівня. По суті, це знакова система, в якій містяться синтаксичні, лексичні і семантичні правила, завдяки чому комп'ютер здійснює необхідні дії. Програмний комплекс «1С» був спеціально розроблений для ведення різного виду бізнесу. Його гнучкість та можливість налаштування «під себе» дає змогу застосовувати його для різних потреб бізнесу, наприклад [41]:

- автоматизації підприємств, що займаються виробництвом і торгівлею, бюджетних і фінансових організацій, підприємств сфери обслуговування тощо; підтримки оперативного управління підприємством;
- автоматизації організаційної та господарської діяльності;
- ведення бухгалтерського обліку з декількома планами рахунків і довільними параметрами обліку, регламентованої звітності;

- вирішення завдань з планування, бюджетування і фінансового аналізу;
- розрахунку заробітної плати та управління персоналом.
- інтелектуальної побудови звітності ієрархічного і багатовимірного типу і крос-звітів;
- збору аналітичної інформації при завданні спеціальних налаштувань без зміни прикладного рішення;
- створення зведених таблиць для аналізу багатовимірної інформації; побудови різного роду діаграм для графічного представлення економічної інформації.

Глобально він нагадує Visual Basic, але зі своїми істотними відмінностями. У нього навіть немає спеціальної назви, а в офіційних джерелах позначений як «Вбудована мова програмування 1С: Підприємство» [42]. Середовище розробки для цієї мови є платформа «1С: Підприємство», а візуальне середовище розробки – це конфігуратор, що є частиною прикладного пакета програм «1С: Підприємство».

До основних переваг платформи «1С» відносяться: вбудована об'єктно-орієнтована мова. З метою полегшення роботи програмістів на платформі «1С: Конфігуратор», розроблені різні допоміжні інструменти (Синтаксис-допомога, колір підсвічування синтаксису, вбудований дебагер тощо); підтримка технології SQL для керування базами даних; представлена можливість створювати індивідуальні проекти, що враховують та відтворюють бізнес-процеси організації; технічна платформа, що є фундаментом для розробки всіх програмних рішень. Такий підхід провокує стандартизацію розробки. Порівняння мов програмування наведено в табл. 3.1.

Таблиця 3.1 – Порівняння характеристик мов програмування

Характеристики	C++	Java	1C
Сфера застосування	Системне програмування, розробка ПЗ, написання драйверів, розважальні ПЗ	Прикладне ПЗ, веб-додатки, розважальні ПЗ, мобільна розробка, клієнтські та серверні додатки	Бізнес-додатки, клієнтські та серверні додатки, веб-додатки
Рік створення	1990 р.	1995 р.	1996 р.
Об'єктно-орієнтована	+	+	+
Імперативна	+	+	+
Функційна	+	-	+
Узагальнена	+	+	+
Процедурна	+	-	-
Рефлексивна	-	+	+
Подійно-орієнтована	-	-	+
Типізація	Статична типізація	Статична типізація	Динамічна типізація
Захищеність типів	-	+	-

Отже, порівнявши ці мови програмування, для розробки інформаційної технології автоматизації роботи супермаркетів було обрано вбудовану мову програмування сімейства програм «1С: Підприємство» – 1С.

3.4 Опис програмних рішень інформаційної технології автоматизації роботи супермаркетів

Щоб на емуляторі екрану вагів створити, кнопки відповідно до тих товарів, які обрав користувач, необхідно, щоб він заповнив відповідні дані, створив

спочатку категорії товарів та обрав в них відповідно товари. Реалізація відбувається поступовим заповненням потрібних даних користувачем.

Діаграма створення товарів на емуляторі екрану вагів зображена на рис. 3.4.



Рисунок 3.4 – Діаграма створення товарів на емуляторі екрану вагів

Зображений вище алгоритм (рис. 3.4) можна записати у вигляді коду:

```

//створення категорій товарів
ХТМЛ = Новый ЗаписьXML;
ХТМЛ.УстановитьСтроку();
ХТМЛ.ЗаписатьНачалоЭлемента( "html");
ХТМЛ.ЗаписатьНачалоЭлемента( "head");
ХТМЛ.ЗаписатьКонецЭлемента(); //head
ХТМЛ.ЗаписатьНачалоЭлемента( "style");
ХТМЛ.ЗаписатьАтрибут("type","text/css");
ХТМЛ.ЗаписатьБезОбработки(
    " a {
    |   display: flex;
    |   justify-content: center;
  
```

```

| align-items: flex-start;
| text-decoration: none;
| color: white;
| font-family: Arial;
| font-size: 20px;
| font-weight: 500;
| border: 3px solid gray;
| background-size: cover;
| background-repeat: no-repeat;
|   background-position: center;
| position: relative;
|   });
XТМЛ.ЗаписатьБезОбработки(
|   .Subgroup2 {
|     display: flex;
|       justify-content: center;
|   }
|     .Subgroup2 a {
|       width: 31vw;
|     }
|   a {
|     height: calc(52vh - 40px);
|     width: 100%;
|   } ");
  XТМЛ.ЗаписатьБезОбработки(
"  .Picture {
|    height: 90%;
|  }
|  a img {
|    width: 100%;
|    height: 100%;
|    object-fit: contain;
|  }
|  span {
|    position: absolute;
|    bottom: 5px;
|    color: black;
|    font-weight: 600;
|  } ");
XТМЛ.ЗаписатьКонецЭлемента(); // style
XТМЛ.ЗаписатьНачалоЭлемента( "body");
XТМЛ.ЗаписатьНачалоЭлемента( "GroupButtons");
Для каждого Строка Из ТаблицаГрупп Цикл
  Если Число(Строка.Номер) = 1 Тогда
    XТМЛ.ЗаписатьНачалоЭлемента( "div");
    XТМЛ.ЗаписатьАтрибут("class","Subgroup1");
  ИначеЕсли Число(Строка.Номер) = 3 И КоличествоГрупп = 4 Тогда
    XТМЛ.ЗаписатьНачалоЭлемента( "div");
    XТМЛ.ЗаписатьАтрибут("class","Subgroup2");
  ИначеЕсли Число(Строка.Номер) = 4 И КоличествоГрупп >= 5 Тогда
    XТМЛ.ЗаписатьНачалоЭлемента( "div");

```

```

        ХТМЛ.ЗаписатьАтрибут("class","Subgroup2");
    КонечЕсли;
    Если Не СтрДлина(Строка.ХранимыйФайл) = 0 Тогда
        Картинка = "data:image/jpg;base64, " + Строка.ХранимыйФайл;
        ХТМЛ.ЗаписатьНачалоЭлемента( "a");
        ХТМЛ.ЗаписатьАтрибут("id","Group" + Строка.Номер);
        ХТМЛ.ЗаписатьАтрибут("href","#");
        ХТМЛ.ЗаписатьНачалоЭлемента("div");
        ХТМЛ.ЗаписатьАтрибут("class","Picture");
        ХТМЛ.ЗаписатьНачалоЭлемента("img");
        ХТМЛ.ЗаписатьАтрибут("src",Картинка);
        ХТМЛ.ЗаписатьКонецЭлемента(); //img
        ХТМЛ.ЗаписатьКонецЭлемента(); //Picture
        ХТМЛ.ЗаписатьНачалоЭлемента("span");
        ХТМЛ.ЗаписатьБезОбработки(Строка.ГруппыТоваров);
        ХТМЛ.ЗаписатьКонецЭлемента(); // span
        ХТМЛ.ЗаписатьКонецЭлемента(); // a
    Иначе
        ХТМЛ.ЗаписатьНачалоЭлемента( "a");
        ХТМЛ.ЗаписатьАтрибут("id","Group" + Строка.Номер);
        ХТМЛ.ЗаписатьАтрибут("href","#");
        ХТМЛ.ЗаписатьНачалоЭлемента("span");
        ХТМЛ.ЗаписатьБезОбработки(Строка.ГруппыТоваров);
        ХТМЛ.ЗаписатьКонецЭлемента(); // span
        ХТМЛ.ЗаписатьКонецЭлемента(); // a
    КонечЕсли;
    Если Число(Строка.Номер) = 2 И КоличествоГрупп = 4 Тогда
        ХТМЛ.ЗаписатьКонецЭлемента(); // Subgroup1
    ИначеЕсли Число(Строка.Номер) = 3 И КоличествоГрупп >= 5 Тогда

        ХТМЛ.ЗаписатьКонецЭлемента(); // Subgroup1
    КонечЕсли;
КонечЦикла;
ХТМЛ.ЗаписатьКонецЭлемента(); // Subgroup2
ХТМЛ.ЗаписатьКонецЭлемента(); //GroupButtons
ХТМЛ.ЗаписатьКонецЭлемента(); //body
ХТМЛ.ЗаписатьКонецЭлемента(); //html
СтрокаХМЛ = ХТМЛ.Закрыть();
Возврат СтрокаХМЛ;
//створення товарів у відповідних категоріях
Если НЕ (Результат = Неопределено) Тогда
    ТаблицаТоваров.Очистить();
    Таблица = Новый ТаблицаЗначений();
    Таблица = (ПолучитьИзВременногоХранилища(Результат));
    Для Индекс = 0 По (Таблица.Количество() -1) Цикл
        Строка = ТаблицаТоваров.Добавить();
        Строка.Номер = Индекс + 1;
        Строка.Код = Строка(Таблица[Индекс].Код);
        Строка.ВаговиеТовари = Таблица[Индекс].ВаговиеТовари;
        Строка.НазваниеГруппы = Таблица[Индекс].НазваниеГруппы;
        Строка.Ссылка = Таблица[Индекс].Ссылка;

```

```

Если Строка.Ссылка.ФайлКартинки.Наименование = "" Тогда
    Строка.ФайлКартинки = Таблица[Индекс].ФайлКартинки;
Иначе
    Строка.ФайлКартинки = Строка.Ссылка.ФайлКартинки;
КонецЕсли;
СтрокаТабГрупп = (ТаблицаГрупп.НайтиСтроки(Новый
Структура("ГруппыТоваров",Строка.НазваниеГруппы))[0]);
Строка.НомерГруппы = (ТаблицаГрупп.Индекс(СтрокаТабГрупп)+1
Строка.КнопкаПосмотреть = "Переглянути";
КонецЦикла;
Для каждого Строка Из ТаблицаГрупп Цикл
    МассивТоваров = ТаблицаТоваров.НайтиСтроки(Новый
Структура("НазваниеГруппы",Строка.ГруппыТоваров));
    Строка.КоличествоТоваров = МассивТоваров.Количество();
КонецЦикла;
КонецЕсли;
XML = Новый ЗаписьXML;
XML.УстановитьСтроку();
XML.ЗаписатьНачалоЭлемента("html");
XML.ЗаписатьНачалоЭлемента("head");
XML.ЗаписатьКонецЭлемента();//head
XML.ЗаписатьНачалоЭлемента("style");
XML.ЗаписатьАтрибут("type","text/css");
Если КолТоваровПоВертикали = 1 Тогда
    XML.ЗаписатьБезОбработки("
        |         body {
        |             margin-top: 12%;
        |             margin-bottom: 12%;
        |         }");
КонецЕсли;
Если НЕ СсылкаТовара = Неопределено Тогда
    ИндексТовара = МассивТоваров.Найти(СсылкаТовара);
КонецЕсли;
Если НЕ ИндексТовара = Неопределено Тогда
    XML.ЗаписатьБезОбработки("
        |         #Product" + Строка(ИндексТовара)+" {" +
        |             border-color: #FFA500;
        |         }");
КонецЕсли;
XML.ЗаписатьКонецЭлемента();//style
XML.ЗаписатьНачалоЭлемента("body");
XML.ЗаписатьНачалоЭлемента("div");
XML.ЗаписатьАтрибут("class","GroupButtons");
Для НомерПодгруппы = 1 По КолТоваровПоВертикали Цикл
    Если Счетчик = СтрокаТабГрупп.КоличествоТоваров ИЛИ Счетчик =
МассивТоваров.Количество() Тогда
        XML.ЗаписатьНачалоЭлемента("div");
        XML.ЗаписатьАтрибут("class","ModelGroup");
        Для МакетКнопки = 1 По КолТовПоГоризонтали Цикл
            XML.ЗаписатьНачалоЭлемента("a");
            XML.ЗаписатьАтрибут("class","ModelButton");

```

```

        ХТМЛ.ЗаписатьАтрибут("href", "#");
        ХТМЛ.ЗаписатьБезОбработки("");
        ХТМЛ.ЗаписатьКонецЭлемента(); //a
    КонецЦикла;
    ХТМЛ.ЗаписатьКонецЭлемента(); //div
    Продолжить;
КонецЕсли;
ХТМЛ.ЗаписатьНачалоЭлемента("div");
ХТМЛ.ЗаписатьАтрибут("class", "Subgroup");
СчетчикПодгруп = СчетчикПодгруп + 1;
Для Индекс = 1 По КолТовПоГоризонтالي Цикл
    Если Счетчик >= СтрокаТабГрупп.КоличествоТоваров ИЛИ Счетчик =
МассивТоваров.Количество() Тогда
        ХТМЛ.ЗаписатьНачалоЭлемента("a");
        ХТМЛ.ЗаписатьАтрибут("class", "ModelButton");
        ХТМЛ.ЗаписатьАтрибут("href", "#");
        ХТМЛ.ЗаписатьБезОбработки("");
        ХТМЛ.ЗаписатьКонецЭлемента(); //a
        Продолжить;
    КонецЕсли;
    КартинкаТовара = ТаблицаТоваров.НайтиСтроки(Новый
Структура("Ссылка", МассивТоваров[Счетчик]))[0].ФайлКартинки;
    Картинка = КартинкаТовара;
    ДвоичныеДанКартинки = ЗагрузкаКартинки(Картинка);
    БинДанные = Base64Строка(ДвоичныеДанКартинки);
    Картинка = "data:image/jpg;base64, " + БинДанные;
    ХТМЛ.ЗаписатьНачалоЭлемента("a");
    ХТМЛ.ЗаписатьАтрибут("id", "Product"+ Строка(Счетчик));
    ХТМЛ.ЗаписатьАтрибут("href", "#");
    ХТМЛ.ЗаписатьНачалоЭлемента("div");
    ХТМЛ.ЗаписатьАтрибут("class", "Picture");
    ХТМЛ.ЗаписатьНачалоЭлемента("img");
    ХТМЛ.ЗаписатьАтрибут("src", Картинка);
    ХТМЛ.ЗаписатьКонецЭлемента(); //img
    ХТМЛ.ЗаписатьКонецЭлемента(); //div
    ХТМЛ.ЗаписатьНачалоЭлемента("span");
ХТМЛ.ЗаписатьКонецЭлемента(); //div
ХТМЛ.ЗаписатьКонецЭлемента(); //body
ХТМЛ.ЗаписатьКонецЭлемента(); //html
СтрокаХМЛ = ХТМЛ.Закрыть();

```

Щоб завантажити на ваги всі товари із зробленими налаштуваннями користувачем, потрібно зібрати всі дані в один файл xml. Діаграма створення файлу для завантаження на ваги зображена на рис. 3.5.

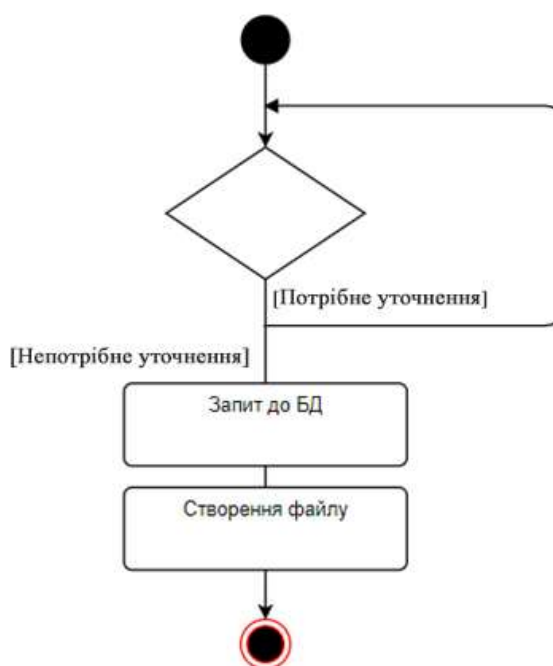


Рисунок 3.5 – Діаграма створення файлу XML для завантаження інформації на ваги

Зображений вище алгоритм (рис. 3.5) можна записати у вигляді коду:

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ТаблицаТоваров.Код,
| ТаблицаТоваров.НомерГруппы,
| ТаблицаТоваров.Ссылка

|ПОМЕСТИТЬ ВТ_ТаблицаТоваров

|ИЗ

| &ТаблицаТоваров КАК ТаблицаТоваров;

////////////////////////////////////

|ВЫБРАТЬ

| ЦеныНоменклатурыСрезПоследних.Номенклатура КАК Номенклатура,
| ЦеныНоменклатурыСрезПоследних.Цена КАК Цена

|ПОМЕСТИТЬ ВТ_Цены

|ИЗ

| РегистрСведений.ЦеныНоменклатуры.СрезПоследних(
| &МоментВремени,
| ВидЦены = &ВидЦены
| И Номенклатура В
| (ВЫБРАТЬ
| Т.Ссылка

| ИЗ

| ВТ_ТаблицаТоваров КАК Т)) КАК ЦеныНоменклатурыСрезПоследних;

```

////////////////////////////////////
|ВЫБРАТЬ
|    КодыТоваровПодключаемогоОборудованияOffline.Номенклатура,
|    КодыТоваровПодключаемогоОборудованияOffline.Код
|ПОМЕСТИТЬ ВТ_КодыОборуд
|ИЗ
|    РегистрСведений.КодыТоваровПодключаемогоОборудованияOffline КАК
КодыТоваровПодключаемогоОборудованияOffline
|ГДЕ
|    КодыТоваровПодключаемогоОборудованияOffline.Номенклатура В
|        (ВЫБРАТЬ
|            Т.Ссылка
|        ИЗ
|            ВТ_ТаблицаТоваров КАК Т)
|    И КодыТоваровПодключаемогоОборудованияOffline.ПравилоОбмена =
&ПравилоОбмена;
////////////////////////////////////
|ВЫБРАТЬ
|    Номенклатура.Ссылка,
|    Номенклатура.ФайлКартинки
|ПОМЕСТИТЬ ВТ_Номенклатура
|ИЗ
|    Справочник.Номенклатура КАК Номенклатура
|ГДЕ
|    Номенклатура.Ссылка В
|        (ВЫБРАТЬ
|            Т.Ссылка
|        ИЗ
|            ВТ_ТаблицаТоваров КАК Т);
////////////////////////////////////
|ВЫБРАТЬ
|    ВТ_КодыОборуд.Код,
|    ВТ_ТаблицаТоваров.НомерГруппы,
|    ВТ_Цены.Цена,
|    ВТ_Номенклатура.Ссылка.НаименованиеПолное КАК Номенклатура,
|    ВТ_Номенклатура.Ссылка.Наименование КАК КраткоеНаименование,
|    ВТ_Номенклатура.Ссылка.ФайлКартинки КАК ФайлКартинки,
|    ВТ_Номенклатура.Ссылка.Описание КАК НоменклатураОписание
|ИЗ
|    ВТ_ТаблицаТоваров КАК ВТ_ТаблицаТоваров
|        ЛЕВОЕ СОЕДИНЕНИЕ ВТ_Цены КАК ВТ_Цены
|        ПО ВТ_ТаблицаТоваров.Ссылка = ВТ_Цены.Номенклатура
|        ЛЕВОЕ СОЕДИНЕНИЕ ВТ_КодыОборуд КАК ВТ_КодыОборуд
|        ПО ВТ_ТаблицаТоваров.Ссылка = ВТ_КодыОборуд.Номенклатура
|        ЛЕВОЕ СОЕДИНЕНИЕ ВТ_Номенклатура КАК ВТ_Номенклатура
|        ПО ВТ_ТаблицаТоваров.Ссылка = ВТ_Номенклатура.Ссылка";
ТЗ = ТаблицаТоваров.Выгрузить(,"код,номергруппы,Ссылка");
Запрос.УстановитьПараметр("ТаблицаТоваров", ТЗ);
Запрос.УстановитьПараметр("ВидЦены", ВидЦены);
Запрос.УстановитьПараметр("МоментВремени", ТекущаяДата());
Запрос.УстановитьПараметр("ПравилоОбмена", ПравилоОбмена);

```

3.5 Тестування роботи програми та аналіз результатів

Розроблена інформаційна технологія для автоматизації роботи супермаркетів була протестована, що підтвердило коректність її роботи.

Було проведено двісті циклів запуску програмного додатку, протестовано можливості його роботи, що дало можливість адекватно оцінити його роботу.

Перш за все користувачу потрібно авторизуватись (рис. 3.6).

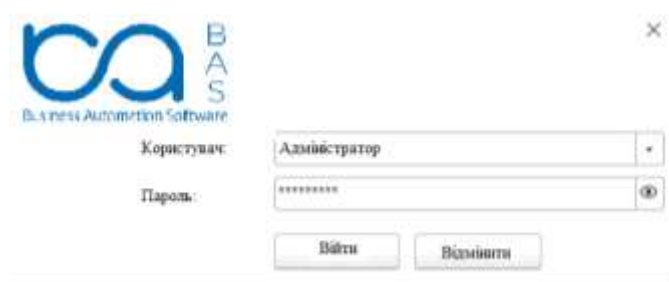


Рисунок 3.6 – Загальний вигляд інтерфейсного вікна авторизації

Після авторизації відкривається вікно загальних налаштувань (рис. 3.7).



Рисунок 3.7 – Загальний вигляд інтерфейсного вікна налаштувань для роботи з вагами

Користувач може ввести кількість категорій товарів, максимальна кількість категорій товарів 6. Якщо користувач введе значення більше 6, то

програма сповістить про помилку та поставить максимально допустиме значення, тобто 6.

Після того як користувач ввів значення кількості категорій товарів, в таблиці з'явилися категорії зі стандартними назвами, які користувач може змінити (рис. 3.8).

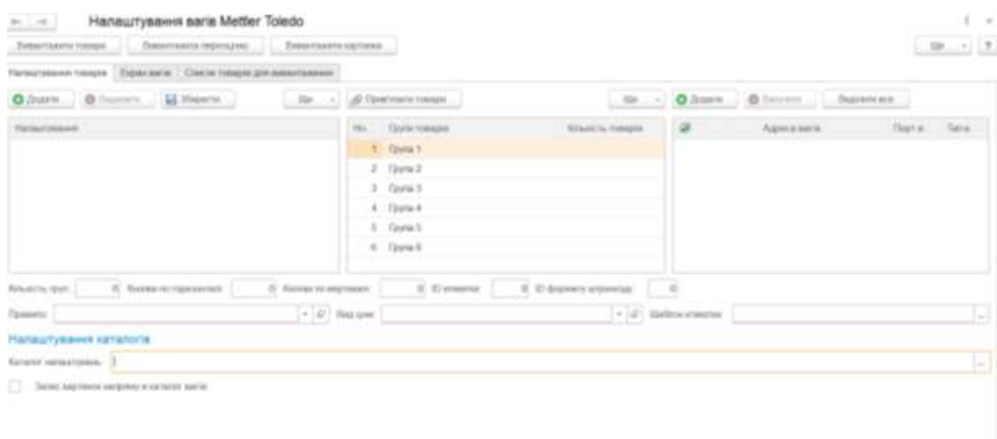


Рисунок 3.8 – Загальний вигляд інтерфейсного вікна створення категорій товарів

Після створення категорій товарів, користувач може відкрити емулятор екрану вагів та побачити їх (рис. 3.9).



Рисунок 3.9 – Загальний вигляд інтерфейсного вікна емулятора екрану вагів

Для того, щоб обрати товари в категорії користувач може натиснути на категорію товарів та в підменю обрати «Обрати товари», або ж відкрити таблицю товарів, яка зображена на рис. 3.11. та натиснути «Додати». Після таких дій відкриється вікно показане на рис. 3.10.



Рисунок 3.10 – Загальний вигляд інтерфейсного вікна з таблицею товарів, що будуть завантажуватись на ваги

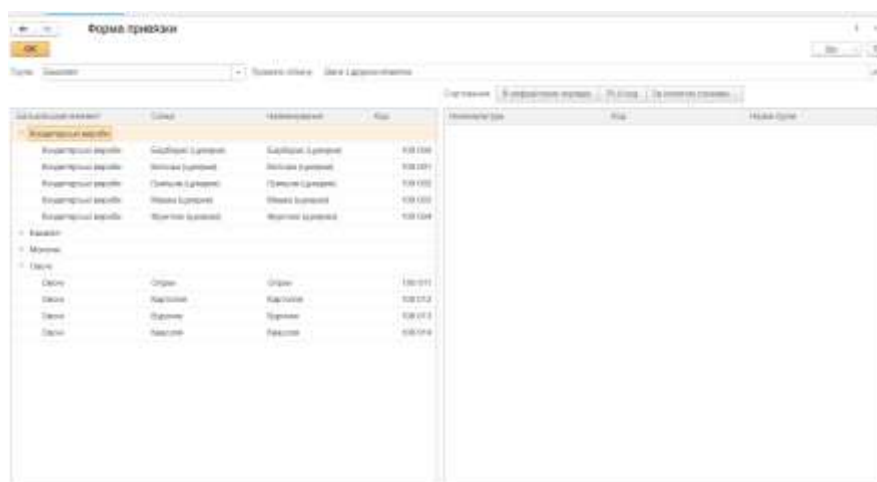


Рисунок 3.11 – Загальний вигляд інтерфейсного вікна роботи з товарами

Зліва є перераховані всі існуючі товари в базі даних. Для того, щоб обрати товари в певну категорію, користувачеві необхідно зверху обрати категорію товару та перенести товар або групу товарів в таблицю справа.

Після таких дій з'являється можливість відсортувати товари за алфавітом, за PLU кодом та за попитом серед споживачів. Після таких дій можна натиснути на кнопку категорії товарів в емуляторі екрану вагів та побачити товари, які були обрані (рис 3.12.).



Рисунок 3.12 – Загальний вигляд інтерфейсного вікна емулятора екрану вагів після додавання товарів в категорію

Ці товари без картинок, але натиснувши на товар, користувач може обрати потрібну картинку (рис. 3.13).

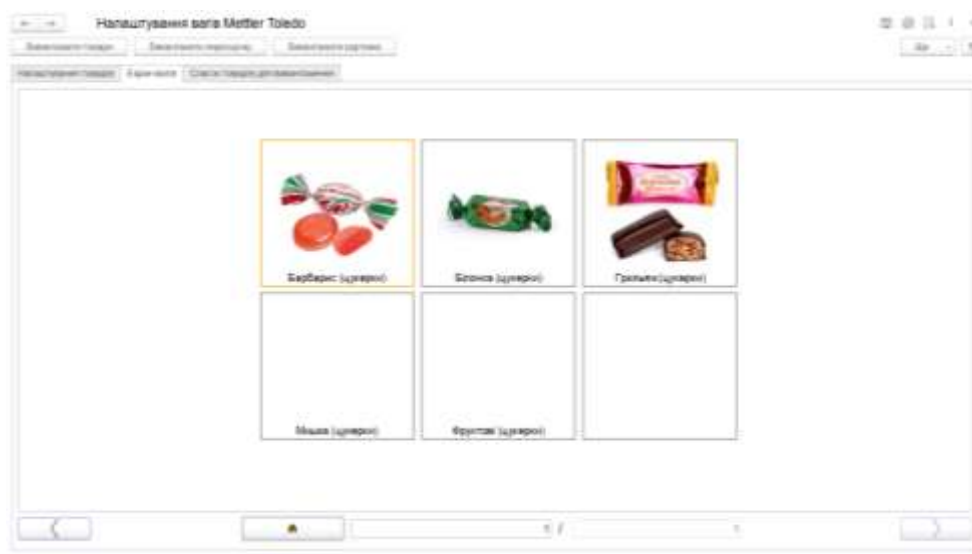
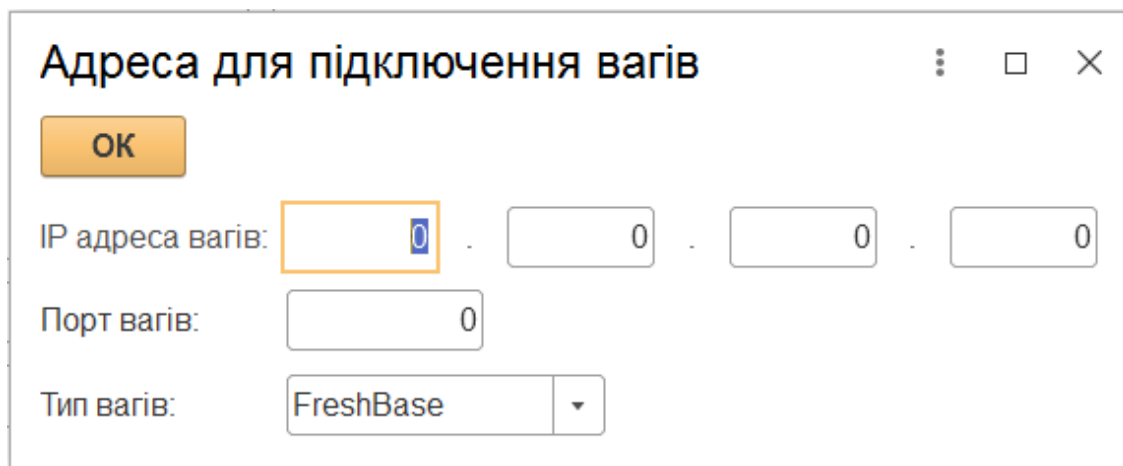


Рисунок 3.13 – Загальний вигляд інтерфейсного вікна емулятора екрану вагів після додавання зображення на товари

Далі користувачеві представлена можливість вказати інформацію про ваги на які буде завантажуватись інформація. Натиснувши на таблицю зі списком вагів, відкривається вікно для введення даних (рис. 3.14).



The image shows a dialog box with the title "Адреса для підключення вагів". It features a yellow "OK" button at the top left. Below it, there are three input fields: "IP адреса вагів:" with four segments (the first containing '0'), "Порт вагів:" with '0', and "Тип вагів:" with a dropdown menu showing "FreshBase".

Рисунок 3.14 – Загальний вигляд інтерфейсного вікна додавання вагів

Після виконання всіх необхідних кроків, можна завантажити інформацію на ваги. Для того, щоб не втратити всі зроблені користувачем налаштування, він може зберегти їх в таблицю налаштувань представлену на головному вікні загальних налаштувань.

У табл. 3.2 наведено порівняльні результати роботи програми з програмами-аналогами: «Формування файлу товарів для завантаження на ваги METTLER TOLEDO Tiger» та «Обробка обслуговування вагів ШТРИХ-М серії РС-200». Розроблена інформаційна технологія має більшу швидкодію та розширений функціонал.

Порівнюючи результати роботи розробленої інформаційної технології з програмами-аналогами, а саме формування файлу товарів для завантаження на ваги «METTLER TOLEDO Tiger» та обробка обслуговування вагів «ШТРИХ-М серії РС-200», можна зробити висновок, що розроблена інформаційна технологія забезпечує високу швидкодію, можливість коригування списку товарів, що

завантажуються на ваги та можливість переглянути вигляд товарів на вагах ще до їх завантаження.

Таблиця 3.2 – Порівняльний аналіз достовірності роботи різних алгоритмів вирішення задачі

Програма	Час завантаження товарів	Можливість відкоригувати список товарів для завантаження	Можливість переглянути вигляд товарів на вагах до їх завантаження
Розроблений додаток	14,19 с.	+	+
Формування файлу товарів для завантаження на ваги METTLER TOLEDO Tiger	20,00 с.	-	-
Обробка обслуговування вагів ШТРИХ-М серії PC-200	21,55 с.	+	-

Швидкодія розробленого програмного забезпечення на 5,81 с більша ніж в програмного забезпечення формування файлу товарів для завантаження на ваги «METTLER TOLEDO Tiger» та на 7,36 с більша ніж в програмної реалізації обслуговування вагів «ШТРИХ-М серії PC-200».

3.6 Висновок до розділу 3

У цьому розділі було розроблено структуру інформаційної технології автоматизації роботи супермаркетів та виокремлено такі модулі: модуль авторизації, модуль загальних налаштувань, модуль роботи з товарами, модуль роботи з вагами, модуль роботи з емулятор сенсорних вагів та модуль роботи з

базою даних. Розроблено схему алгоритму та описано детально його кроки, щоб мати набір послідовних інструкцій для реалізації програмного забезпечення.

Було проаналізовано та порівняно мови програмування C++, Java та 1С. Для розробки інформаційної технології автоматизації роботи супермаркетів було обрано вбудовану мову програмування сімейства програм «1С: Підприємство» – 1С, оскільки вона була спеціально розроблена для ведення середнього або малого бізнесу, має динамічну типізацію та подійно-орієнтованість.

Проведено тестування роботи інформаційної технології автоматизації роботи супермаркетів та проаналізовано її результати. Тестування програми підтвердило очікувані результати та правильність роботи програми. Розроблена інформаційна технологія забезпечує високу швидкість, можливість коригування списку товарів, що завантажуються на ваги та можливість переглянути вигляд товарів на вагах ще до їх завантаження. Швидкість розробленого програмного продукту на 5,81 с більша ніж в додатку формування файлу товарів для завантаження на ваги «METTLER TOLEDO Tiger» та на 7,36 с більша ніж в додатку обробки обслуговування вагів «ШТРИХ-М серії РС-200».

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку нового програмного продукту інформаційної технології автоматизації роботи супермаркетів». Інформаційна технологія, що допомагає автоматизувати налаштування та обміни товарами між програмним забезпеченням супермаркету та сенсорними вагами самообслуговування. Основною перевагою є можливість відсортувати товари за потрібними показниками, наприклад:

- якщо потрібно продати товар, який дуже довго стоїть на складі,
- якщо потрібно продати товар на який є знижка чи акція,
- якщо потрібно продати сезонний товар.

Аналогом може бути ШТРИХ-М серії РС-200, ціна 25 000 грн.; Mettler toledo , ціна 15 000 грн.; Штрих-ПРИНТ 4.5, ціна 18 000 грн. Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 4.1 [43].

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах

Продовження табл. 4.1

Бали (за 5-ти бальною шкалою)					
Кри- терій	0	1	2	3	4
Ринкові переваги					
2	Багато аналогів на малому ринку	Ринкові п Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно до-рівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі вла-стивості продук-ту значно гірші, ніж в аналогів	Технічні та споживчі вла-стивості продук-ту трохи гірші, ніж в аналогів	Технічні та споживчі вла-стивості продук-ту на рівні аналогів	Технічні та споживчі вла-стивості продук-ту трохи кращі, ніж в аналогів	Технічні та споживчі вла-стивості про-дукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуата-ційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позити-вної міки	Ринок малий, але має пози-тивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих ком-паній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з ко-мерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне не-значне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так із комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фі-нансування ідеї відсутні	Потрібні незначні фі-нансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фі-нансування є	Потрібні незначні фінансові ресурси. Джерела фі-нансування є	Не потребує додаткового фінансування

Продовження табл. 4.1

Бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в табл. 4.2

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	4	4	4
Наявність аналогів на ринку	4	3	3

Продовження табл. 4.2

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Цінова політика	4	4	4
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	3	3	4
Ринок збуту	4	3	3
Конкурентоспроможність	3	4	4
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	3	4	3
Матеріально-технічна база	4	3	3
Термін реалізації ідеї	4	4	3
Супровідна документація	4	3	4
Сума	45	41	43
Середньоарифметична сума балів	$(45+41+43) / 3 = 43$		

За даними табл. 4.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в табл. 4.3.

Таблиця 4.3 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ , розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок того, що інформаційна технологія відрізняється від існуючих тим, що за рахунок автоматизації поточних операцій (налаштування та завантаження товарів на сенсорні ваги самообслуговування) спрощує роботу персоналу та збільшує ефективність роботи супермаркетів і магазинів. Основною перевагою є можливість відсортувати товари за потрібними показниками, наприклад:

- якщо потрібно продати товар, який дуже довго стоїть на складі,
- якщо потрібно продати товар на який є знижка чи акція,
- якщо потрібно продати сезонний товар.

4.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

Основна заробітна плата розробників, яка розраховується за формулою [44]:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (4.1)$$

де M – місячний посадовий оклад конкретного розробника (дослідника), грн.;

T_p – число робочих днів в місяці, 21 днів;

t – число днів роботи розробника (дослідника).

Результати розрахунків зведемо до табл. 4.4.

Таблиця 4.4 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	40000	1904,76	35	66666,667
Інженер	30000	1428,57	35	50000,000
Всього				116666,67

Оскільки в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

Додаткова заробітна плата прийнято розраховувати як 11 % від основної заробітної плати розробників та робітників:

$$Z_d = Z_o \cdot 11 \% / 100 \%, \quad (4.2)$$

$$Z_d = (116666,67 \cdot 11 \% / 100 \%) = 12833,33 \text{ (грн.)}.$$

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$H_z = (Z_o + Z_d) \cdot 22 \% / 100\%, \quad (4.3)$$

$$H_z = (116666,67 + 12833,33) \cdot 22 \% / 100 \% = 28490,00 \text{ (грн.)}.$$

Оскільки для розроблювального пристрою не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді амортизація обладнання, що використовувалась для розробки розраховується за формулою:

$$A = \frac{Ц}{T \cdot 12} \cdot t_{вик} \text{ [грн.]}, \quad (4.4)$$

де $Ц$ – балансова вартість обладнання, грн.;

T – термін корисного використання обладнання згідно податкового законодавства, років;

$t_{вик}$ – термін використання під час розробки, місяців.

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова

вартість якого становить 20500 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 1,67 міс.

$$A_{обл} = \frac{20500}{2} \times \frac{1,67}{12} = 1423,611(\text{грн}).$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до табл. 4.5. Для розрахунку амортизації нематеріальних ресурсів використовується формула:

$$A_{н.р.} = Ц_{н.р.} * H_a * \frac{t_{вик.}}{12}. \quad (4.5)$$

Але, так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів менше 20000 грн, то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю, $V_{нем.ак.} = 13600$ грн.

Таблиця 4.5 – Амортизаційні відрахування матеріальних і нематеріальних ресурсів для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та компютерна периферія	20500	2	1,67	1423,611
Офісне обладнення	22000	4	1,67	763,889
Приміщення	1100000	20	1,67	7638,889
Всього				9826,39

Тарифи на електроенергію для побутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При

цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot P \cdot \Phi \cdot K_n, \quad (4.6)$$

де V – вартість 1 кВт-години електроенергії для 1 класу підприємства,
 $V = 6,2$ грн./кВт;

P – встановлена потужність обладнання, кВт. $P = 0,6$ кВт;

Φ – фактична кількість годин роботи обладнання, годин;

K_n – коефіцієнт використання потужності, $K_n = 0,9$.

$$V_e = 0,9 \cdot 0,6 \cdot 8 \cdot 35 \cdot 6,2 = 937,44 \text{ (грн.)}.$$

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{iv}}{100\%}, \quad (4.7)$$

де H_{iv} – норма нарахування за статтею «Інші витрати».

$$I_e = 116666,67 * 130\% / 100\% = 151666,7 \text{ (грн.)}.$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати,

пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.8)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{нзв} = 116666,67 * 110 \% / 100 \% = 128333 \text{ (грн.)}.$$

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{заг} = 116666,67 + 12833,33 + 28490,00 + 9826,39 + 13600 + 937,44 + 151666,7 + 128333 = 462353,83 \text{ (грн.)}.$$

Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються ZB , визначається за формулою:

$$ZB = \frac{B_{заг}}{\eta} \text{ [грн]}, \quad (4.9)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta=0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$. Оберемо $\eta = 0,5$, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ЗВ = 462353,83 / 0,5 = 924708 \text{ (грн.)}.$$

4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки [45].

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

- 1) вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;
- 2) зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного

інвестора (наприклад, протягом 3-х років після її впровадження);

3) кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;

4) визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем. В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta\Pi_0 \cdot N + \Pi_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (4.10)$$

де $\pm \Delta\Pi_0$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

C_o – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $C_o = C_0 \pm \Delta C_o$;

C_b – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$;

p – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2022 році $\vartheta = 18\%$.

Припустимо, що при прогнозованій ціні 5000 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 500 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 1000 шт., протягом другого року – на 2000 шт., протягом третього року на 3000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\begin{aligned} \Delta\Pi_1 &= (0*500 + (5000 + 500)*1000)*0,8333*0,5*(1 - 0,18) = \\ &= 1708333,265 \text{ (грн.)}, \end{aligned}$$

$$\begin{aligned} \Delta\Pi_2 &= (0*500 + (5000 + 500)*(1000+2000))*0,8333*0,5*(1 - 0,18) = \\ &= 5637499,775 \text{ (грн.)}, \end{aligned}$$

$$\begin{aligned} \Delta\Pi_3 &= (0*500 + (5000 + 500)*(1000+2000+3000))*0,8333*0,5*(1 - 0,18) = \\ &= 11274999,549 \text{ (грн.)}. \end{aligned}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 18620832,59 грн.

Розрахунок ефективності вкладених інвестицій та періоду їх окупності.
Розраховуємо приведену вартість збільшення всіх чистих прибутків $ПП$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (4.11)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

T – період часу, протягом якого виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t – період часу (в роках).

Збільшення прибутку ми отримаємо починаючи з першого року:

$$\begin{aligned} ПП &= (1708333,265 / (1+0,1)^1) + (5637499,775 / (1+0,1)^2) + (11274999,549 / (1+0,1)^3) = \\ &= 1553030,24 + 4659090,723 + 8471074,041 = 14683195 \text{ (грн.)}. \end{aligned}$$

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{инв} * ЗВ, \quad (4.12)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку

приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{инв}=2...5$, але може бути і більшим;

ZB – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 924708 = 1849415,32 \text{ (грн.)}$$

Тоді абсолютний економічний ефект E_{abc} або чистий приведений дохід (NPV , *Net Present Value*) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{abc} = ПП - PV, \quad (4.13)$$

$$E_{abc} = 14683195 - 1849415,32 = 12833779,69 \text{ (грн.)}$$

Оскільки $E_{abc} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (IRR , *Internal Rate of Return*) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_e . Для цього використаємо формулу:

$$E_e = T_{ж} \sqrt{1 + \frac{E_{abc}}{PV}} - 1, \quad (4.14)$$

де $T_{жк}$ – життєвий цикл наукової розробки, роки.

$$E_g = \sqrt[3]{(1 + 12833779,69/1849415,32) - 1} = 0,995.$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (4.15)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні $d = (0,09 \dots 0,14)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05 \dots 0,5)$.

$$\tau_{\min} = 0,14 + 0,05 = 0,19.$$

Оскільки $E_b > \tau_{\min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_g}, \quad (4.16)$$

$$T_{ок} = 1 / 0,995 = 1,005 \text{ (р.)}.$$

Оскільки $T_{ок} < 3$ -х років, а саме термін окупності рівний 1,005 роки, то фінансування даної наукової розробки є доцільним.

4.4 Висновок до розділу 4

Економічна частина роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 924708 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 1,005 роки.

ВИСНОВКИ

У ході виконання магістерської кваліфікаційної роботи реалізовано інформаційну технологію автоматизації роботи супермаркетів.

Проведено огляд систем-аналогів, які створені для автоматизації завантаження товарів на сенсорні ваги самообслуговування, коротко охарактеризовано їх головні функції та наведено основні переваги та недоліки. Виявлено проблему відсутності емулятора сенсорних вагів самообслуговування та можливості сортувати товари за різних потреб.

Було розроблено структуру інформаційної технології автоматизації роботи супермаркетів та спроектовано схему алгоритму роботи розроблюваного програмного забезпечення. Розглянуто структурну організацію системи завантаження товарів та інформації про них на сенсорні ваги.

На основі аналізу існуючих моделей та методів, які застосовуються для вирішення поставленої задачі було обґрунтовано використання в інформаційній технології методу ранжування, який є основою розробленої математичної моделі формування списку товарів для завантаження на сенсорні ваги. Така модель підвищить ефективність формування списку товарів та прискорить швидкодію процесу обмінів між інформаційною базою та вагами самообслуговування.

Здійснено проектування та розробку інформаційної технології автоматизації роботи супермаркетів. Обґрунтовано вибір мови програмування. Розробка інформаційної технології велася на об'єктно-орієнтованій мові програмування 1С. Здійснено програмну реалізацію інформаційної технології автоматизації роботи супермаркетів, детально описано класи та функції, які використовуються в інформаційній технології.

Проведено тестування роботи інформаційної технології автоматизації роботи супермаркетів та проаналізовано її результати. Тестування програми підтвердило очікувані результати та правильність роботи програми, а саме можливість переглянути вигляд товарів на вагах ще до їх завантаження.

Швидкодія розробленого програмного продукту на 5,81 с більша ніж в додатку формування файлу товарів для завантаження на ваги «METTLER TOLEDO Tiger» та на 7,36 с більша ніж в додатку обробки обслуговування вагів «ШТРИХ-М серії РС-200», що означає доведення поставленої мети.

Здійснено економічне обґрунтування доцільності розробки інформаційної технології автоматизації роботи супермаркетів.

Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, знайдено термін окупності витрат для виробника та економічний ефект для споживача при використанні такої розробки. В результаті аналізу розрахунків можна зробити висновок, що розробка у виробництві та використання дешевша за аналог і є висококонкурентоспроможною. Період окупності складе близько 1,005 роки. Загальні витрати становлять 924708 грн.

Завдання, які були поставлені для реалізації інформаційної технології автоматизації роботи супермаркетів, виконані в повному обсязі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Буксир Я. О., Крилик Л. В. Розробка програмного додатку для автоматизації роботи супермаркетів // *Матеріали І науково-технічної конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2021) : збірник доповідей* [Електронний ресурс]. Вінниця: ВНТУ, 2021. С. 551–552.

URL: https://conferences.vntu.edu.ua/public/files/1/vntu_2021_netpub.pdf. (дата звернення: 18.11.22).

2. Буксир Я. О., Крилик Л. В. Перспективи розробки інформаційної технології автоматизації роботи супермаркетів // *Матеріали ІІ науково-технічної конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2022) : збірник доповідей* [Електронний ресурс]. Вінниця: ВНТУ, 2022. С. 720–722 (PDF, 2830 с.).

URL: https://conferences.vntu.edu.ua/public/files/1/vntu_2022_netpub.pdf. (дата звернення: 18.11.22).

3. Комп'ютерна програма «Інформаційна технологія автоматизації роботи супермаркетів» : свідоцтво про реєстрацію авторського права на твір №114837, видане ДП «Український інститут інтелектуальної власності» (Укрпатент) від 16.09.2022 р. / Крилик Л. В., Буксир Я. О.

4. Супермаркет [Електронний ресурс].
URL: <https://web.archive.org/web/20190209125959/https://zakon.rada.gov.ua/laws/show/z1350-05> (дата звернення: 02.09.2022).

5. Сенсорні ваги самообслуговування [Електронний ресурс].
URL: <https://systemgroup.com.ua/uk/equipment/vagove-obladnannya/vagy-samoobslugovuvannya> (дата звернення: 02.09.2022).

6. Що таке мова XML [Електронний ресурс]. URL: <https://serpstat.com/uk/blog/chto-takoe-jazyk-xml/> (дата звернення: 10.09.2022).

7. Програмне забезпечення для ведення бізнесу [Електронний ресурс]. URL: https://aktiv.ua/bas_ (дата звернення: 10.09.2022).
8. BAS [Електронний ресурс]. URL: <https://www.bas-https://uit.kiev.ua/bas-chi-1s-pidpriemstvo/> (дата звернення: 15.09.2022).
9. Різновиди BAS [Електронний ресурс]. URL: <https://www.netsoft.com.ua/articles-soft/bas-news/chto-takoje-business-automation-software-i-jego-raznovidnosti.html> (дата звернення: 15.09.2022).
10. Завантаження товарів на ваги Штрих-ПРИНТ 4.5 [Електронний ресурс]. URL: <https://infostart/public/949785/>. (дата звернення: 20.09.2022).
11. Обробка обслуговування вагів з друком етикеток [Електронний ресурс]. URL: <http://tf21/public/1162896/>. (дата звернення: 20.09.2022).
12. Завантаження товарів на ваги Штрих принт Ф1 TOUCH [Електронний ресурс]. URL: <https://infostart/public/1067072/>. (дата звернення: 20.09.2022).
13. Бідюк П.І., Терентьев О.М., Коновалюк М.М. Байєсівські мережі в технологіях інтелектуального аналізу даних. *Штучний інтелект*. 2010. № 2. С. 104–113. URL: <http://dspace.nbuv.gov.ua/bitstream/handle/123456789/56141/13-Bidyuk.pdf?sequence=1> (дата звернення: 20.09.2022).
14. Прикладна статистика та основи економетрики [Електронний ресурс]. URL: <https://ukrbukva.net/page,2,43106-Prikladnaya-statistika-i-osnovy-ekonometriki.html> (дата звернення: 20.09.2022).
15. Вступ до Баєсових мереж [Електронний ресурс]. URL: <https://lib.chmnu.edu.ua/pdf/posibnuku/313/10.pdf> (дата звернення: 20.09.2022).
16. Методи обробки експертної інформації [Електронний ресурс]. URL: <https://www.wiki.uk-ua.nina.az> (дата звернення: 22.09.2022).
17. Прийняття рішень з використанням експертних оцінок [Електронний ресурс]. URL: <https://learn.ztu.edu.ua/pluginfile.php/145514> (дата звернення: 23.09.2022).

18. Як продавати продукти в торгівельних мережах [Електронний ресурс]. URL: <https://leosvit.com/art/yak-prodavaty-produkt-v-torgovelnih-merezhah> (дата звернення: 25.09.2022).
19. Метод ранжування [Електронний ресурс]. URL: https://stud.com.ua/50988/menedzhment/kilkisni_metodi_otsinki (дата звернення: 28.09.2022).
20. UML [Електронний ресурс]. URL: <https://www.uml.org/> (дата звернення: 5.10.2022).
21. Побудова UML [Електронний ресурс]. URL: https://studwood.net/1598583/informatika/pobudova_diagrami_osnovi_proektu_bazi_danih_kasa_prodazhu_kvitkiv (дата звернення: 5.10.2022).
22. Діаграма класів [Електронний ресурс]. URL: <https://naurok.com.ua/prezentaciya-diagrami-uml-diagrami-klasiv> (дата звернення: 8.10.2022).
23. Реляційні бази даних [Електронний ресурс]. URL: https://pns.hneu.edu.ua/pluginfile.php/601514/mod_resource/content/ (дата звернення: 10.10.2022).
24. Основні поняття реляційних БД: нормалізація, зв'язок та ключі [Електронний ресурс]. URL: <https://bondarenko.dn.ua/osnovni-ponyattya-relyatsijnih-bd-normalizatsiya-zv-yazok-ta-klyuchi/> (дата звернення: 10.10.2022).
25. ER-модель. Поняття зв'язку [Електронний ресурс]. URL: <https://www.bestprog.net/uk/2019/01/27/er-model-the-concept-of-relationship-the-relationship-capacity-types-of-relationships-examples-ua/> (дата звернення: 10.10.2022).
26. Інформаційні бази [Електронний ресурс]. URL: https://its.1c/db/metod8dev/content/1591/hdoc_/ (дата звернення: 15.10.2022).
27. Робота з базою даних [Електронний ресурс]. URL: <https://v8.1c/platforma/rabota-s-bazoy-dannykh/> (дата звернення: 15.10.2022).

28. Довідники [Електронний ресурс]. URL: https://sitis.com.ua/ua/about/articles/spravochniki_1s/(дата звернення: 20.10.2022).
29. Реєстри відомостей [Електронний ресурс]. URL: <https://www.ateasyday.com/articles/programs/kak-dobavit-ili-udalit-zapis-v-registre-svedenij-1s.html> (дата звернення: 20.10.2022).
30. Реєстри накопичення [Електронний ресурс]. URL: <https://www.ua.koderline.com.ua/expert/instruktsii/article-instruktsiya-stvorennya-zvitu-v-1s-8-3-erp/> (дата звернення: 20.10.2022).
31. Реєстри бухгалтерії [Електронний ресурс]. URL: <https://trend1c.com/blog/registry-bukhgalterii-1s/> (дата звернення: 20.10.2022).
32. Реєстри розрахунку [Електронний ресурс]. URL: <https://i.factor.ua/ukr/journals/bk/2010/february/issue-4/article-100728>. (дата звернення: 20.10.2022).
33. 1С:Підприємство [Електронний ресурс]. URL: <https://uit.kiev.ua/baschi-1s-pidpriemstvo/>. (дата звернення: 25.10.2022).
34. Що таке платформа? [Електронний ресурс]. URL: <https://tqm.com.ua/ua/likbez/chary/sho-take-platforma-1c>. (дата звернення: 25.10.2022).
35. Що таке конфігурація? [Електронний ресурс]. URL: <https://kamala-soft.com/uk/blog/konfiguratsii-1s-pidpriemstvo-vidy-i-versii-programm/> (дата звернення: 25.10.2022).
36. Архітектура 1С Підприємство [Електронний ресурс]. URL: <https://v8.1c.ru/metod/article/arkhitektura-1s-predpriyatiya-kak-produkt-inzhenernoj-mysli.htm>. (дата звернення: 28.10.2022).
37. Поняття алгоритму [Електронний ресурс]. URL: <https://naurok.com.ua/test/ponyattya-algoritmu-vlastivosti-algoritmiv-sposobi-podannya-algoritmiv-774595.html> (дата звернення: 02.11.2022).
38. Огляд і основи мови програмування C++ [Електронний ресурс]. URL: http://www.znannya.org/?view=Cpp_basics_ (дата звернення: 08.11.2022).

39. Можливості та технології Java [Електронний ресурс]. URL: <https://studway.com.ua/code-part-6/>. (дата звернення: 08.11.2022).
40. Переваги мови програмування Java [Електронний ресурс]. URL: <https://qagroup.com.ua/publications/benefits-of-java/> (дата звернення: 08.11.2022).
41. Переваги мови програмування 1С [Електронний ресурс]. URL: <https://unipro.com.ua/ua/perevagy-ta-nedoliky-programy-kompaniyi--1s----1s--buhgalteriya--1s--pidpryyemstvoredpryyatyue/> (дата звернення: 08.11.2022).
42. 1С: Підприємство [Електронний ресурс]. URL: <https://kamala-soft.com/blog/yazyk-programirovaniya-1s-pidpriemstvo/> (дата звернення: 15.11.2022).
43. Технологічний аудит [Електронний ресурс]. URL: <https://studfile.net/preview/7336428/> (дата звернення: 25.11.22).
44. Прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської) та конструкторсько-технологічної роботи [Електронний ресурс]. URL: <https://web.posibnyku.vntu.edu.ua>. (дата звернення: 25.11.22).
45. Ефективність наукових досліджень [Електронний ресурс]. URL: <https://studfile.net/preview/6872280/page:25/>. (дата звернення: 27.11.22).

Додаток А (обов'язковий) Результат перевірки на плагіат в онлайн-системі UNICHECK



Имя пользователя:
Озеранський В.С. КН

ID проверки:
1013269622

Дата проверки:
11.12.2022 18:09:00 EET

Тип проверки:
Doc vs Internet + Library

Дата отчета:
11.12.2022 18:11:51 EET

ID пользователя:
62038

Название файла: 122МКР-БуксирЯО2022

Количество страниц: 68 Количество слов: 11850 Количество символов: 94327 Размер файла: 1.07 MB ID файла: 1013028396

11.8% Совпадения

Наибольшее совпадение: 2.6% с источником из Библиотеки (ID файла: 1013028393)

5.47% Источники из Интернета 7 Страница 70

7.99% Источники из Библиотеки 121 Страница 70

0% Цитат

Исключение цитат выключено

Исключение списка библиографических ссылок выключено

40% Исключений

Некоторые источники исключены автоматически (фильтры исключения: количество найденных слов меньш...

6.12% Исключений из Интернета 71 Страница 71

35.5% Исключенного текста из Библиотеки 206 Страница 71

Модификации

Обнаружены модификации текста. Подробная информация доступна в онлайн-отчете.

Замененные символы 50

Додаток Б (обов'язковий) Лістинг програми

```
//Створення категорій товарів
Функция СоздатьКнопкиГрупп(ИндексГруппы)
    ХТМЛ = Новый ЗаписьXML;
    ХТМЛ.УстановитьСтроку();
    ХТМЛ.ЗаписатьНачалоЭлемента( "html");
    ХТМЛ.ЗаписатьНачалоЭлемента( "head");
    ХТМЛ.ЗаписатьКонецЭлемента(); //head
    ХТМЛ.ЗаписатьНачалоЭлемента( "style");
    ХТМЛ.ЗаписатьАтрибут("type","text/css");
    ХТМЛ.ЗаписатьБезОбработки(
        " a {
        |   display: flex;
        |   justify-content: center;
        |   align-items: flex-start;
        |   text-decoration: none;
        |   color: white;
        |   font-family: Arial;
        |   font-size: 20px;
        |   font-weight: 500;
        |   border: 3px solid gray;
        |   background-size: cover;
        |   background-repeat: no-repeat;
        |       background-position: center;
        |   position: relative;
        |   }
        |   .Subgroup1,
        |   .Subgroup2 {
        |       display: flex;
        |           justify-content: center;
        |   }
        |   .Picture {
        |       height: 90%;
        |   }
        |   a img {
        |       width: 100%;
        |       height: 100%;
        |       object-fit: contain;
        |   }
        |   span {
        |       position: absolute;
        |       bottom: 5px;
        |       color: black;
        |       font-size: 24px;
        |       font-weight: 600;
```

```

|   }");
Если НЕ ИндексГруппы = Неопределено Тогда
  HTML.ЗаписатьБезОбработки("
  "      #Group" + Строка(ИндексГруппы)+" {"      +
  "
  |           border-color: #FFA500;
  |   }");
КонецЕсли;
HTML.ЗаписатьКонецЭлемента(); // style
HTML.ЗаписатьНачалоЭлемента( "body");
HTML.ЗаписатьНачалоЭлемента( "GroupButtons");
Для каждого Строка Из ТаблицаГрупп Цикл
  Если Число(Строка.Номер) = 1 Тогда
    HTML.ЗаписатьНачалоЭлемента( "div");
    HTML.ЗаписатьАтрибут("class", "Subgroup1");
  ИначеЕсли Число(Строка.Номер) = 3 И КоличествоГрупп = 4 Тогда
    HTML.ЗаписатьНачалоЭлемента( "div");
    HTML.ЗаписатьАтрибут("class", "Subgroup2");
  ИначеЕсли Число(Строка.Номер) = 4 И КоличествоГрупп >= 5 Тогда
    HTML.ЗаписатьНачалоЭлемента( "div");
    HTML.ЗаписатьАтрибут("class", "Subgroup2");
  КонецЕсли;

Если Не СтрДлина(Строка.ХранимыйФайл) = 0 Тогда
  Картинка = "data:image/jpeg;base64, " + Строка.ХранимыйФайл;
  HTML.ЗаписатьНачалоЭлемента( "a");
  HTML.ЗаписатьАтрибут("id", "Group" + Строка.Номер);
  HTML.ЗаписатьАтрибут("href", "#");
  HTML.ЗаписатьНачалоЭлемента("div");
  HTML.ЗаписатьАтрибут("class", "Picture");
  HTML.ЗаписатьНачалоЭлемента("img");
  HTML.ЗаписатьАтрибут("src", Картинка);
  HTML.ЗаписатьКонецЭлемента(); //img
  HTML.ЗаписатьКонецЭлемента(); //Picture
  HTML.ЗаписатьНачалоЭлемента("span");
  HTML.ЗаписатьБезОбработки(Строка.ГруппыТоваров);
  HTML.ЗаписатьКонецЭлемента(); // span
  HTML.ЗаписатьКонецЭлемента(); // a
Иначе
  HTML.ЗаписатьНачалоЭлемента( "a");
  HTML.ЗаписатьАтрибут("id", "Group" + Строка.Номер);
  HTML.ЗаписатьАтрибут("href", "#");
  HTML.ЗаписатьНачалоЭлемента("span");
  HTML.ЗаписатьБезОбработки(Строка.ГруппыТоваров);
  HTML.ЗаписатьКонецЭлемента(); // span
  HTML.ЗаписатьКонецЭлемента(); // a
КонецЕсли;
Если Число(Строка.Номер) = 2 И КоличествоГрупп = 4 Тогда

```



```

        ХТМЛ.ЗаписатьКонецЭлемента(); // Subgroup1
    ИначеЕсли Число(Строка.Номер) = 3 И КоличествоГрупп >= 5 Тогда
        ХТМЛ.ЗаписатьКонецЭлемента(); // Subgroup1
    КонецЕсли;
КонецЦикла;
ХТМЛ.ЗаписатьКонецЭлемента(); // Subgroup2
ХТМЛ.ЗаписатьКонецЭлемента(); //GroupButtons
ХТМЛ.ЗаписатьКонецЭлемента(); //body
ХТМЛ.ЗаписатьКонецЭлемента(); //html
СтрокаХМЛ = ХТМЛ.Закрыть();
Возврат СтрокаХМЛ;
КонецФункции // ()

//Створення товарів
&НаСервере
Функция СоздатьКнопкиТоваровХТМЛ(ИндексГруппы, ИндексТовара, СсылкаТовара)
    СтрокаТабГрупп = ТаблицаГрупп.НайтиСтроки(Новый
Структура("Номер", Число(ИндексГруппы)))[0];
    //розрахунок кількості кнопок
    КолТовНаСтранице = КолТовПоГоризонтали * КолТоваровПоВертикали;
    КолСтраниц = СтрокаТабГрупп.КоличествоТоваров / КолТовНаСтранице;
    НеПолнаяЗакладка = СтрокаТабГрупп.КоличествоТоваров % КолТовНаСтранице;
    Если НеПолнаяЗакладка > 0 Тогда
        КолСтраниц = Цел(КолСтраниц + 1);
    КонецЕсли;
    СуммаСтраниц = КолСтраниц;
    Если СуммаСтраниц = 1 И ТекущаяСтраница = 1 Тогда
        Элементы.КомандаВперед.Доступность = Ложь;
        Элементы.КомандаНазад.Доступность = Ложь;
    ИначеЕсли (НЕ СуммаСтраниц = 1) И ТекущаяСтраница = 1 Тогда
        Элементы.КомандаВперед.Доступность = Истина;
        Элементы.КомандаНазад.Доступность = Ложь;
    Иначе
        Элементы.КомандаВперед.Доступность = Истина;
        Элементы.КомандаНазад.Доступность = Истина;
    КонецЕсли;
    МассивТоваровГруппы = ТаблицаТоваров.НайтиСтроки(Новый
Структура("НазваниеГруппы", СтрокаТабГрупп.ГруппыТоваров));
    ///Поділ товарів кожної сторінки на масиви.
    МассивСтраниц = Новый Массив;
    Индекс = 0;
    Для НомерСтраницы = 1 По КолСтраниц Цикл
        МассивТоваровСтраницы = Новый Массив;
        Если Индекс = МассивТоваровГруппы.Количество() Тогда
            Прервать;
        КонецЕсли;
        Для НомерСтроки = 0 По КолТовНаСтранице - 1
            Цикл//МассивТоваров.Количество() Цикл

```

```

        Если Индекс = МассивТоваровГруппы.Количество() Тогда
            Прервать;
        КонецЕсли;
МассивТоваровСтраници.Добавить(МассивТоваровГруппы[Индекс].Ссылка);
        Индекс = Индекс + 1;
        КонецЦикла;
        МассивСтраниц.Добавить(МассивТоваровСтраници);
КонецЦикла;
Счетчик = 0;
СчетчикПодгруп = 1;
МассивТоваров = МассивСтраниц[ТекущаяСтраница-1];
ХТМЛ = Новый ЗаписьXML;
ХТМЛ.УстановитьСтроку();
ХТМЛ.ЗаписатьНачалоЭлемента("html");
ХТМЛ.ЗаписатьНачалоЭлемента("head");
ХТМЛ.ЗаписатьКонецЭлемента(); //head
ХТМЛ.ЗаписатьНачалоЭлемента("style");
ХТМЛ.ЗаписатьАтрибут("type","text/css");
ХТМЛ.ЗаписатьБезОбработки("
|     body {
|         margin-top: 2%;
|         margin-bottom: 5%;
|     }");
ХТМЛ.ЗаписатьБезОбработки("
|     .Subgroup {
|         display: flex;
|         justify-content: center;
|     }
|     a {
|         display: flex;
|         justify-content: center;
|         align-items: flex-end;
|         margin: 4px;
|         text-decoration: none;
|         font-family: 'Arial';
|         font-size:15px;
|         font-weight: 500;
|         border: 2px solid gray;
|         background-size: cover;
|         background-repeat: no-repeat;
|         background-position: center;
|         color: #000000;
|         position: relative;
|     }
|     .ModelButton{
|         border: 2px solid gray;
|         margin: 4px;
|     }
|     }

```

```

|     .Picture {
|         width: 99%;
|         height: 95%;
|     });
HTML.ЗаписатьБезОбработки("
|     a {
|         height: 200px;
|         width: 100%;
|     }
|     .ModelButton{
|         height: 200px;
|         width: 100%;
|     }
|     .Picture img {
|         width: 100%;
|         height: 90%;
|         object-fit: contain;
|     }
|     span {
|         position: absolute;
|         bottom: 2px;
|     }
|     .ModelGroup {
|         display: flex;
|         justify-content: center;
| });
Если НЕ СсылкаТовара = Неопределено Тогда
    ИндексТовара = МассивТоваров.Найти(СсылкаТовара);
КонецЕсли;
Если НЕ ИндексТовара = Неопределено Тогда
    HTML.ЗаписатьБезОбработки("
        "     #Product" + Строка(ИндексТовара)+" {" +
        "
        |         border-color: #FFA500;
        |     });
КонецЕсли;
HTML.ЗаписатьКонецЭлемента(); //style
HTML.ЗаписатьНачалоЭлемента("body");
HTML.ЗаписатьНачалоЭлемента("div");
HTML.ЗаписатьАтрибут("class","GroupButtons");
Для НомерПодгруппы = 1 По КолТоваровПоВертикали Цикл
    Если Счетчик = СтрокаТабГрупп.КоличествоТоваров ИЛИ Счетчик =
МассивТоваров.Количество() Тогда
        HTML.ЗаписатьНачалоЭлемента("div");
        HTML.ЗаписатьАтрибут("class","ModelGroup");
        Для МакетКнопки = 1 По КолТовПоГоризонтали Цикл
            HTML.ЗаписатьНачалоЭлемента("a");

```

```

        HTML.ЗаписатьАтрибут("class","ModelButton");
        HTML.ЗаписатьАтрибут("href","#");
        HTML.ЗаписатьБезОбработки("");
        HTML.ЗаписатьКонецЭлемента();//а
    КонецЦикла;
    HTML.ЗаписатьКонецЭлемента();//div
    Продолжить;
КонецЕсли;
HTML.ЗаписатьНачалоЭлемента("div");
HTML.ЗаписатьАтрибут("class","Subgroup");
СчетчикПодгруп = СчетчикПодгруп + 1;
Для Индекс = 1 По КолТовПоГоризонтали Цикл
    Если Счетчик >= СтрокаТабГрупп.КоличествоТоваров ИЛИ Счетчик =
МассивТоваров.Количество() Тогда
        HTML.ЗаписатьНачалоЭлемента("a");
        HTML.ЗаписатьАтрибут("class","ModelButton");
        HTML.ЗаписатьАтрибут("href","#");
        HTML.ЗаписатьБезОбработки("");
        HTML.ЗаписатьКонецЭлемента();//а
        Продолжить;
    КонецЕсли;
    КартинкаТовара = ТаблицаТоваров.НайтиСтроки(Новый
Структура("Ссылка",МассивТоваров[Счетчик]))[0].ФайлКартинки;
    Если КартинкаТовара =
Справочники.НоменклатураПрисоединенныеФайлы.ПустаяСсылка() Тогда
        HTML.ЗаписатьНачалоЭлемента("a");
        HTML.ЗаписатьАтрибут("id","Product" + Строка(Счетчик));
        HTML.ЗаписатьАтрибут("href","#");
        HTML.ЗаписатьНачалоЭлемента("span");
        HTML.ЗаписатьБезОбработки(МассивТоваров[Счетчик].Наименование);
        HTML.ЗаписатьКонецЭлемента();//span
        HTML.ЗаписатьКонецЭлемента();//а
    Иначе
        Картинка = КартинкаТовара;
        ДвоичныеДанКартинки = ЗагрузкаКартинки(Картинка);
        БинДанные = Base64Строка(ДвоичныеДанКартинки);
        Картинка = "data:image/jpg;base64, " + БинДанные;
        HTML.ЗаписатьНачалоЭлемента("a");
        HTML.ЗаписатьАтрибут("id","Product"+ Строка(Счетчик));
        HTML.ЗаписатьАтрибут("href","#");
        HTML.ЗаписатьНачалоЭлемента("div");
        HTML.ЗаписатьАтрибут("class","Picture");
        HTML.ЗаписатьНачалоЭлемента("img");
        HTML.ЗаписатьАтрибут("src", Картинка);
        HTML.ЗаписатьКонецЭлемента();//img
        HTML.ЗаписатьКонецЭлемента();//div
        HTML.ЗаписатьНачалоЭлемента("span");

```

```

ХТМЛ.ЗаписатьБезОбработки(МассивТоваров[Счетчик].Наименование);
        ХТМЛ.ЗаписатьКонецЭлемента(); //span
        ХТМЛ.ЗаписатьКонецЭлемента(); // a
        КонецЕсли;
        Счетчик = Счетчик + 1;
        КонецЦикла;
        ХТМЛ.ЗаписатьКонецЭлемента(); //div
КонецЦикла;
ХТМЛ.ЗаписатьКонецЭлемента(); //div
ХТМЛ.ЗаписатьКонецЭлемента(); //body
ХТМЛ.ЗаписатьКонецЭлемента(); //html
СтрокаХМЛ = ХТМЛ.Закрыть();
Возврат СтрокаХМЛ;
КонецФункции // ()

//Відкриття вікна для роботи з товарами
&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
    Адрес = Параметры.Адрес;
    СтруктураПараметров = ПолучитьИзВременногоХранилища(Адрес);
    ТаблицаТоваров.Загрузить(СтруктураПараметров.ТабТоваров);
    ТаблицаГрупп = СтруктураПараметров.ТабГрупп;
    ПравилоОбмена = СтруктураПараметров.ПравилоОбмена;
    ТекущаяГруппаТоваров = СтруктураПараметров.ТекущаяСтрокаТаб;
    Для Индекс = 0 По (ТаблицаГрупп.Количество()-1) Цикл
        ЭтаФорма.Элементы.Группа.СписокВыбора.Добавить(ТаблицаГрупп[Индекс].ГруппыТо
варов);
        КонецЦикла;

        Если Не ТекущаяГруппаТоваров = Неопределено Тогда
            Группа = ТекущаяГруппаТоваров;
        ИначеЕсли ТаблицаГрупп.Количество() > 0 Тогда
            Группа = ТаблицаГрупп[0].ГруппыТоваров;
        КонецЕсли;
        Зарос = Новый Запрос;
        Запрос.Текст =
            "ВЫБРАТЬ
            |     КодыТоваровПодключаемогоОборудованияOffline.Номенклатура.Ссылка КАК
Ссылка,
            |     КодыТоваровПодключаемогоОборудованияOffline.Код КАК Код,
            |
            |     КодыТоваровПодключаемогоОборудованияOffline.Номенклатура.Наименование КАК
Наименование,
            |     КодыТоваровПодключаемогоОборудованияOffline.Номенклатура.Родитель
КАК Родитель,
            |     ВЫБОР

```

```

|          КОГДА
КодыТоваровПодключаемогоОборудованияOffline.Номенклатура.ЕстьТоварыДругогоКачеств
а
|          ТОГДА 4 + ВЫБОР
|          КОГДА
КодыТоваровПодключаемогоОборудованияOffline.Номенклатура.ПометкаУдаления
|          ТОГДА 1
|          ИНАЧЕ 0
|          КОНЕЦ + ВЫБОР
|          КОГДА
КодыТоваровПодключаемогоОборудованияOffline.Номенклатура.ИспользованиеХарактерис
тик =
ЗНАЧЕНИЕ(Перечисление.ВариантыИспользованияХарактеристикНоменклатуры.НеИспользуй
вать)
|          ТОГДА 0
|          ИНАЧЕ 2
|          КОНЕЦ
|          ИНАЧЕ ВЫБОР
|          КОГДА
КодыТоваровПодключаемогоОборудованияOffline.Номенклатура.ПометкаУдаления
|          ТОГДА 1
|          ИНАЧЕ 0
|          КОНЕЦ + ВЫБОР
|          КОГДА
КодыТоваровПодключаемогоОборудованияOffline.Номенклатура.ИспользованиеХарактерис
тик =
ЗНАЧЕНИЕ(Перечисление.ВариантыИспользованияХарактеристикНоменклатуры.НеИспользуй
вать)
|          ТОГДА 0
|          ИНАЧЕ 2
|          КОНЕЦ
|          КОНЕЦ КАК ИндексКартинки
| ИЗ
| РегистрСведений.КодыТоваровПодключаемогоОборудованияOffline КАК
КодыТоваровПодключаемогоОборудованияOffline
| ГДЕ
| КодыТоваровПодключаемогоОборудованияOffline.ПравилоОбмена =
&ПравилоОбмена
| ИТОГИ ПО
| Родитель";

Запрос.УстановитьПараметр("ПравилоОбмена", ПравилоОбмена);
ПромДерево =
Запрос.Выполнить().Выгрузить(ОбходРезультатаЗапроса.ПоГруппировкамСИерархией);
ЗначениеВРеквизитФормы(ПромДерево, "ДеревоНоменклатуры");
КонецПроцедуры
//Перетаскування товарів в таблицю для завантаження
&НаСервере

```

Процедура ПереборЭлементовДерева(ТекущееЗначение)

```

    Запрос = Новый Запрос;
    Запрос.Текст =
    "ВЫБРАТЬ
    |     КодыТоваровПодключаемогоОборудованияOffline.Номенклатура.Ссылка КАК
Ссылка,
    |     КодыТоваровПодключаемогоОборудованияOffline.Код КАК Код
    | ИЗ
    |     РегистрСведений.КодыТоваровПодключаемогоОборудованияOffline КАК
КодыТоваровПодключаемогоОборудованияOffline
    | ГДЕ
    |     КодыТоваровПодключаемогоОборудованияOffline.Номенклатура.Ссылка В
ИЕРАРХИИ (&ТекущееЗначение)";

```

```

    Запрос.УстановитьПараметр("ТекущееЗначение", ТекущееЗначение);
    РезультатЗапроса = Запрос.Выполнить().Выбрать();
    Пока РезультатЗапроса.Следующий() Цикл
        Если РезультатЗапроса.Ссылка.ЭтоГруппа = Ложь Тогда
            Если ТаблицаТоваров.НайтиСтроки(Новый Структура("Ссылка"
,РезультатЗапроса.Ссылка)).Количество() > 0 Тогда
                Продолжить;
            Иначе
                НоваяСтрока = ТаблицаТоваров.Добавить();
                НоваяСтрока.Ссылка = РезультатЗапроса.Ссылка;
                НоваяСтрока.Ваговиетовари = НоваяСтрока.Ссылка.Наименование;
                НоваяСтрока.Код = РезультатЗапроса.Код;
                НоваяСтрока.НазваниеГруппы = Группа;
                КонецЕсли;
            КонецЕсли;
        КонецЦикла;
    КонецПроцедуры

```

//Завантаження товарів на ваги

&НаКлиенте

Процедура ЗаписьXMLНаКлиенте()

```

    Счетчик = 1;
    ВыгрузитьТоварыНаСервере();
    ИмяФайла = КаталогНастроек + "\xml\PLUData.xml";
    ЗаписьXML = Новый ЗаписьXML;
    ЗаписьXML.ОткрытьФайл(ИмяФайла, "UTF-8"); //
    ЗаписьXML.ЗаписатьОбъявлениеXML();

    ЗаписьXML.ЗаписатьНачалоЭлемента("Message");
    ЗаписьXML.ЗаписатьНачалоЭлемента("ARTSCommonHeader");
    ЗаписьXML.ЗаписатьАтрибут("MessageType", "Request");
    ЗаписьXML.ЗаписатьКонецЭлемента();
    ЗаписьXML.ЗаписатьНачалоЭлемента("ItemTransaction");
    ЗаписьXML.ЗаписатьАтрибут("ActionCode", "Write");

```

```

        Для каждого Выборка Из ТаблицаЗаписьXML Цикл
            PLU = ПолучитьПЛУ(Выборка.Код);
            ЗаписьXML.ЗаписатьНачалоЭлемента("Item");
                ЗаписатьXML(ЗаписьXML, PLU,"PLU");
                ЗаписьXML.ЗаписатьНачалоЭлемента("Descriptions");
//описание                ЗаписьXML.ЗаписатьАтрибут("Action","Create");
                ЗаписьXML.ЗаписатьНачалоЭлемента("Description");
                ЗаписьXML.ЗаписатьАтрибут("Type","ItemName");
                ЗаписьXML.ЗаписатьТекст(Выборка.Номенклатура);
                ЗаписьXML.ЗаписатьКонецЭлемента();
                ЗаписьXML.ЗаписатьНачалоЭлемента("Description");
                ЗаписьXML.ЗаписатьАтрибут("Type","ItemShortName");
            ЗаписьXML.ЗаписатьТекст(Выборка.КраткоеНаименование);
        ЗаписьXML.ЗаписатьКонецЭлемента();
            ЗаписьXML.ЗаписатьНачалоЭлемента("Description");
            ЗаписьXML.ЗаписатьАтрибут("Type","ExtraText");
            ЗаписьXML.ЗаписатьТекст(СокрЛП(Выборка.НоменклатураОписание))
            ЗаписьXML.ЗаписатьКонецЭлемента();
                ЗаписьXML.ЗаписатьНачалоЭлемента("AlternativeItemIDs");
            ЗаписьXML.ЗаписатьАтрибут("Action","Create");
        ЗаписатьXML(ЗаписьXML, Формат(Выборка.Код,"ЧГ=0"),"AlternativeItemID");
            ЗаписьXML.ЗаписатьКонецЭлемента();
            ЗаписьXML.ЗаписатьНачалоЭлемента("ItemPrices"); //цена
            ЗаписьXML.ЗаписатьАтрибут("Action","Create");
            ЗаписьXML.ЗаписатьНачалоЭлемента("ItemPrice");
            ЗаписьXML.ЗаписатьАтрибут("ValueTypeCode","BasePrice")
            ЗаписьXML.ЗаписатьАтрибут("UnitOfMeasureCode","KGM");           Если не
        ЗначениеЗаполнено(Выборка.Цена) или Выборка.Цена = Неопределено тогда
            ЗаписьXML.ЗаписатьТекст("0");
        Иначе
            ЗаписьXML.ЗаписатьТекст(Формат(Выборка.Цена,"ЧДЦ=2; ЧРД="));
        КонецЕсли;
            ЗаписьXML.ЗаписатьКонецЭлемента();
            ЗаписьXML.ЗаписатьНачалоЭлемента("LabelFormats"); //этикетка
            ЗаписьXML.ЗаписатьАтрибут("Action","Create");
            ЗаписьXML.ЗаписатьНачалоЭлемента("LabelFormatID");
            ЗаписьXML.ЗаписатьАтрибут("Index","0");
        Если ЗначениеЗаполнено(ИдЭтикетки) Тогда
            ЗаписьXML.ЗаписатьТекст(Строка(ИдЭтикетки));
        Иначе
            ЗаписьXML.ЗаписатьТекст("1");
        КонецЕсли;
            ЗаписьXML.ЗаписатьКонецЭлемента();
            ЗаписьXML.ЗаписатьКонецЭлемента();
            ЗаписьXML.ЗаписатьНачалоЭлемента("Barcodes"); //штрихкод
            ЗаписьXML.ЗаписатьАтрибут("Action","Create");
            Если ЗначениеЗаполнено(ИдФорматаШтрихкода) Тогда

```



```

        ЗаписатьXML(ЗаписьXML,
Строка(ИдФорматаШтрихкода),"BarcodeID");
        Иначе
            ЗаписатьXML(ЗаписьXML, "1","BarcodeID");
        КонецЕсли;
ЗаписьXML.ЗаписатьКонецЭлемента();
Если ЗначениеЗаполнено(Выборка.ФайлКартинки) Тогда
ЗаписьXML.ЗаписатьНачалоЭлемента("ImageNames"); //Картинка
ЗаписьXML.ЗаписатьАтрибут("Action","Create");
    //<ImageNames Action="Create">
ЗаписьXML.ЗаписатьНачалоЭлемента("ImageName");
        //<ImageName Type="Item">apple1</ImageName>
        ЗаписьXML.ЗаписатьАтрибут("Type","Item");
ЗаписьXML.ЗаписатьТекст(Формат(PLU,"ЧГ=0"));
        ЗаписьXML.ЗаписатьКонецЭлемента();
        ЗаписьXML.ЗаписатьКонецЭлемента();
    КонецЕсли;
ЗаписьXML.ЗаписатьКонецЭлемента(); //item

    Прогресс = (Счетчик * 100)/ТаблицаЗаписьXML.Количество();
    Состояние("Залишилось:", Прогресс);
    Счетчик = Счетчик +1;
КонецЦикла;
ЗаписьXML.ЗаписатьКонецЭлемента();
ЗаписьXML.ЗаписатьКонецЭлемента();
ЗаписьXML.Закреть();
ТекстДок = Новый ТекстовыйДокумент;
ТекстДок.Прочитать(ИмяФайла,,);
ТекстДок.Записать(ИмяФайла,"UTF-8",Символы.ПС);

КонецПроцедур

```

Додаток В (обов'язковий)

107


ІЛЮСТРАТИВНА ЧАСТИНА
«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АВТОМАТИЗАЦІЇ
РОБОТИ СУПЕРМАРКЕТІВ»

Виконала: студентка 2-го курсу, групи
ІКН-21м

Спеціальності 122 – «Комп'ютерні
науки»


Буксир Я. О.
(прізвище та ініціали)

Керівник: д.т.н., доц. каф. КН


Крилик Л. В.
(прізвище та ініціали)
« 15 » 12 . 2022 р.

Вінниця ВНТУ - 2022 рік

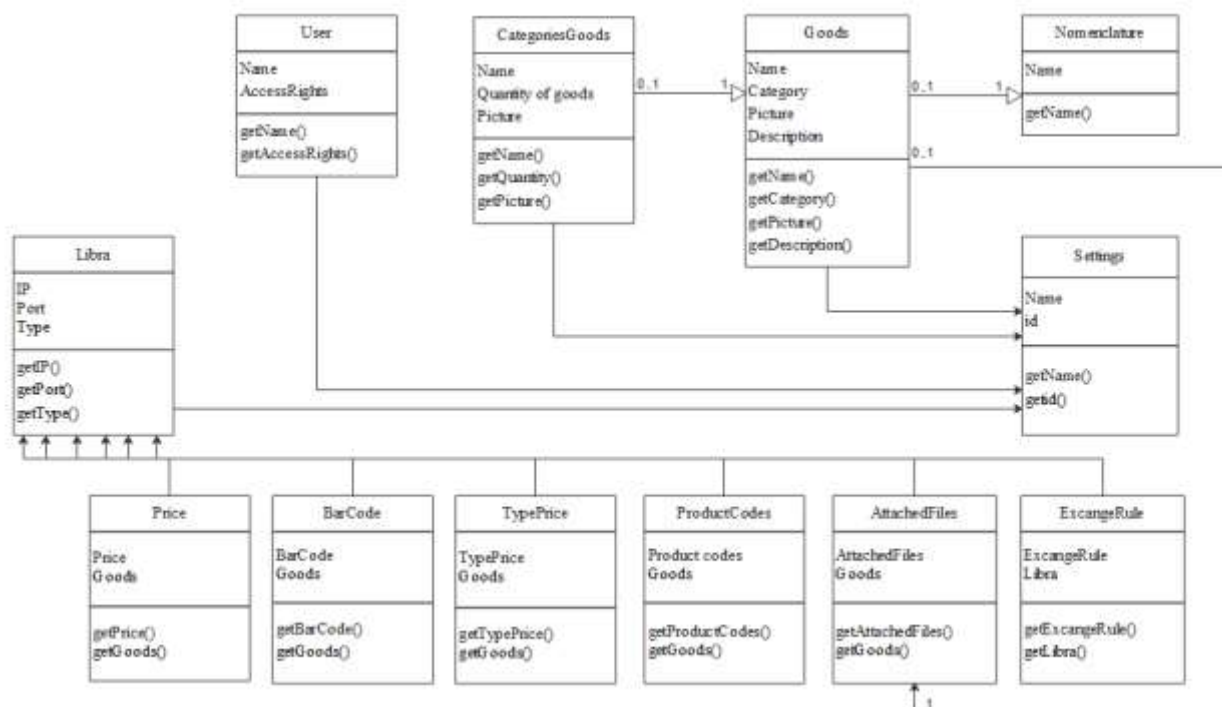


Рисунок В.1 – UML-діаграма асоціацій класів інформаційної технології автоматизації роботи супермаркетів

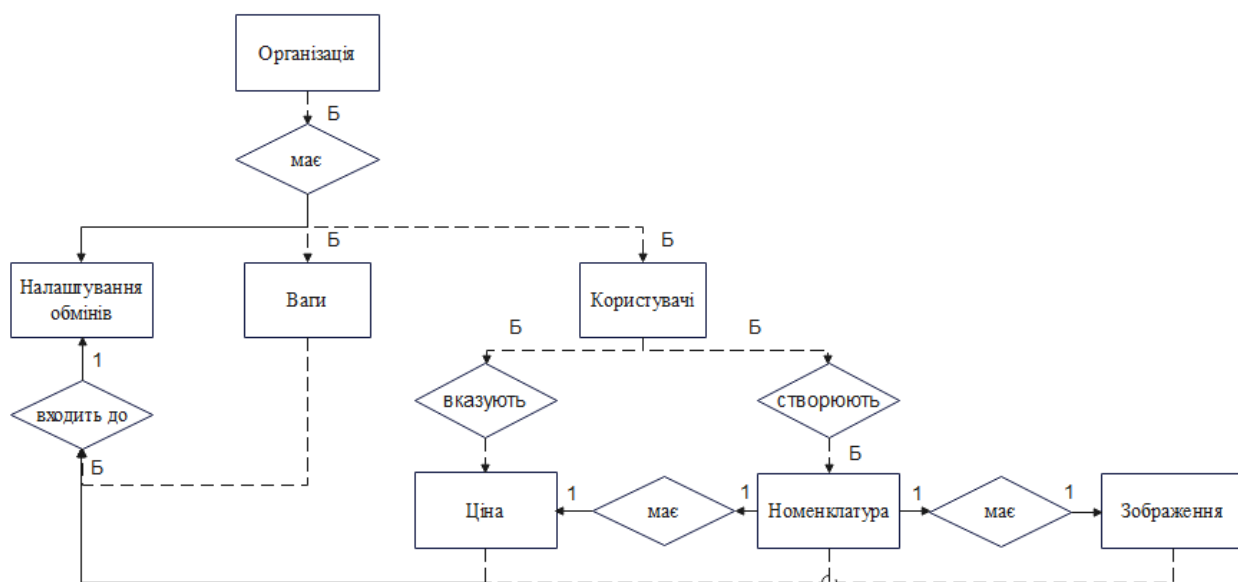


Рисунок В.2 – Схема ER-моделі бази даних інформаційної технології автоматизації роботи супермаркетів

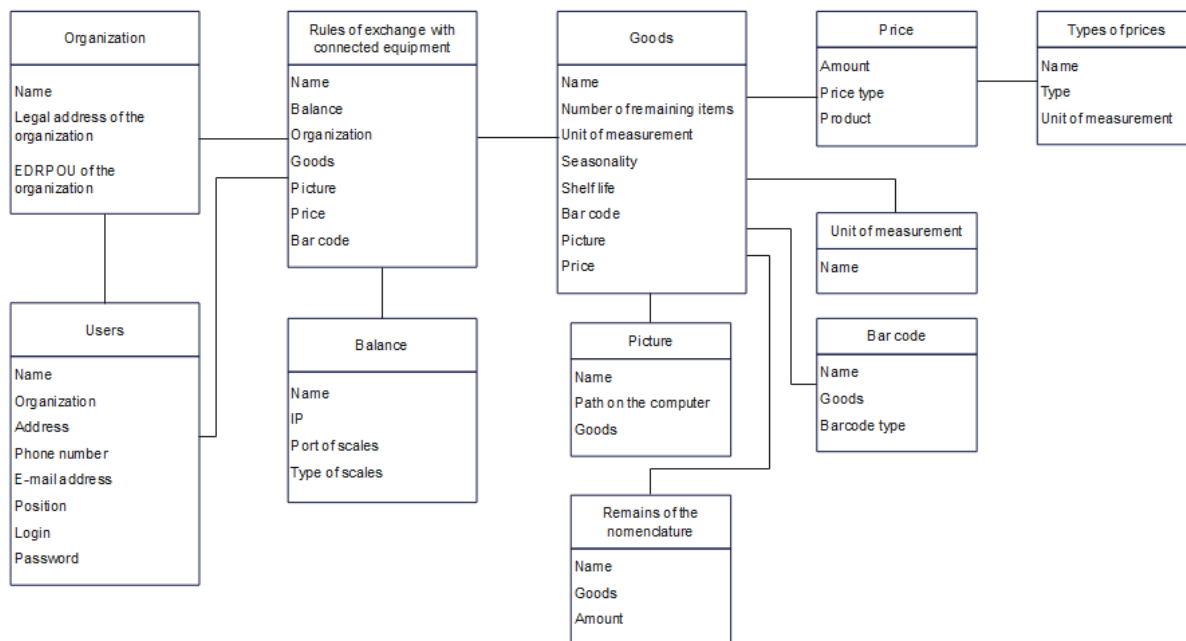


Рисунок В.3 – Схема зв'язків у базі даних інформаційної технології автоматизації роботи супермаркетів

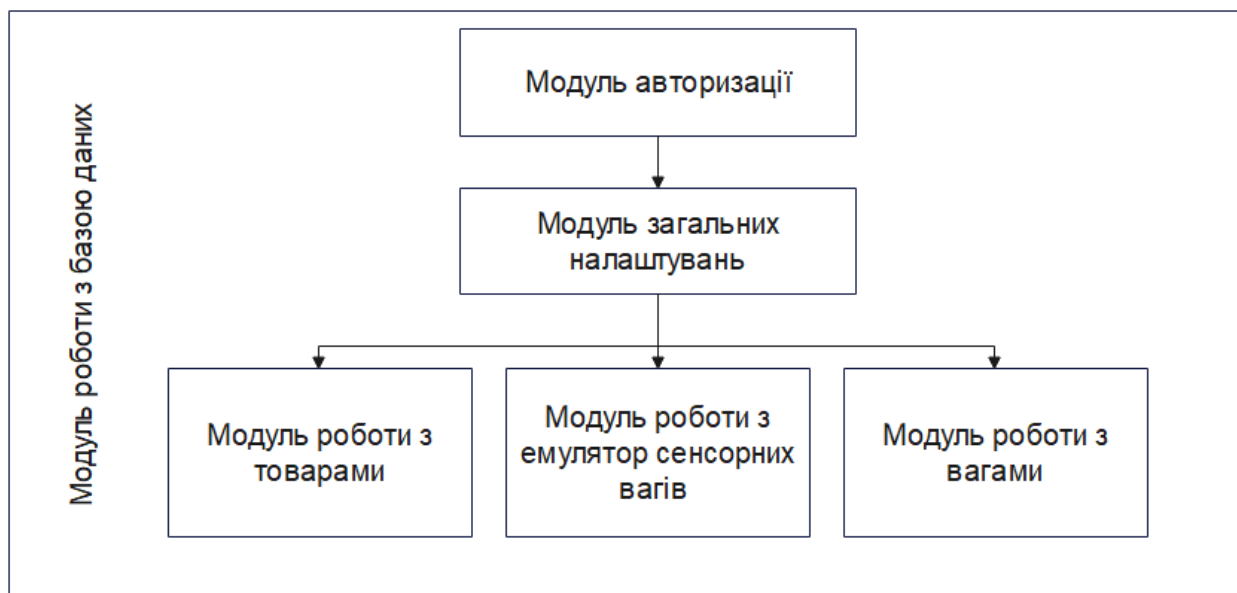


Рисунок В.4 – Структурна схема інформаційної технології автоматизації роботи супермаркетів

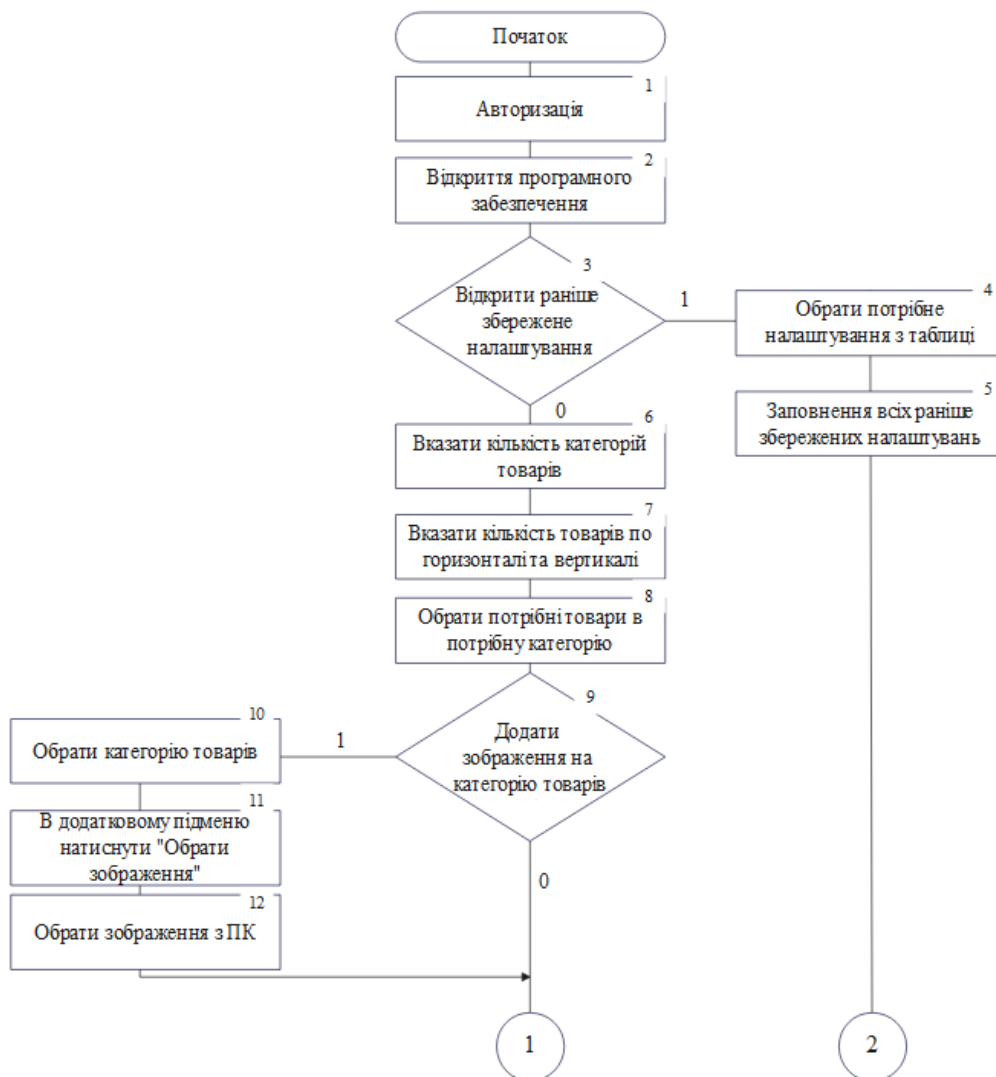


Рисунок В.5 – Схема загального алгоритму функціонування інформаційної технології

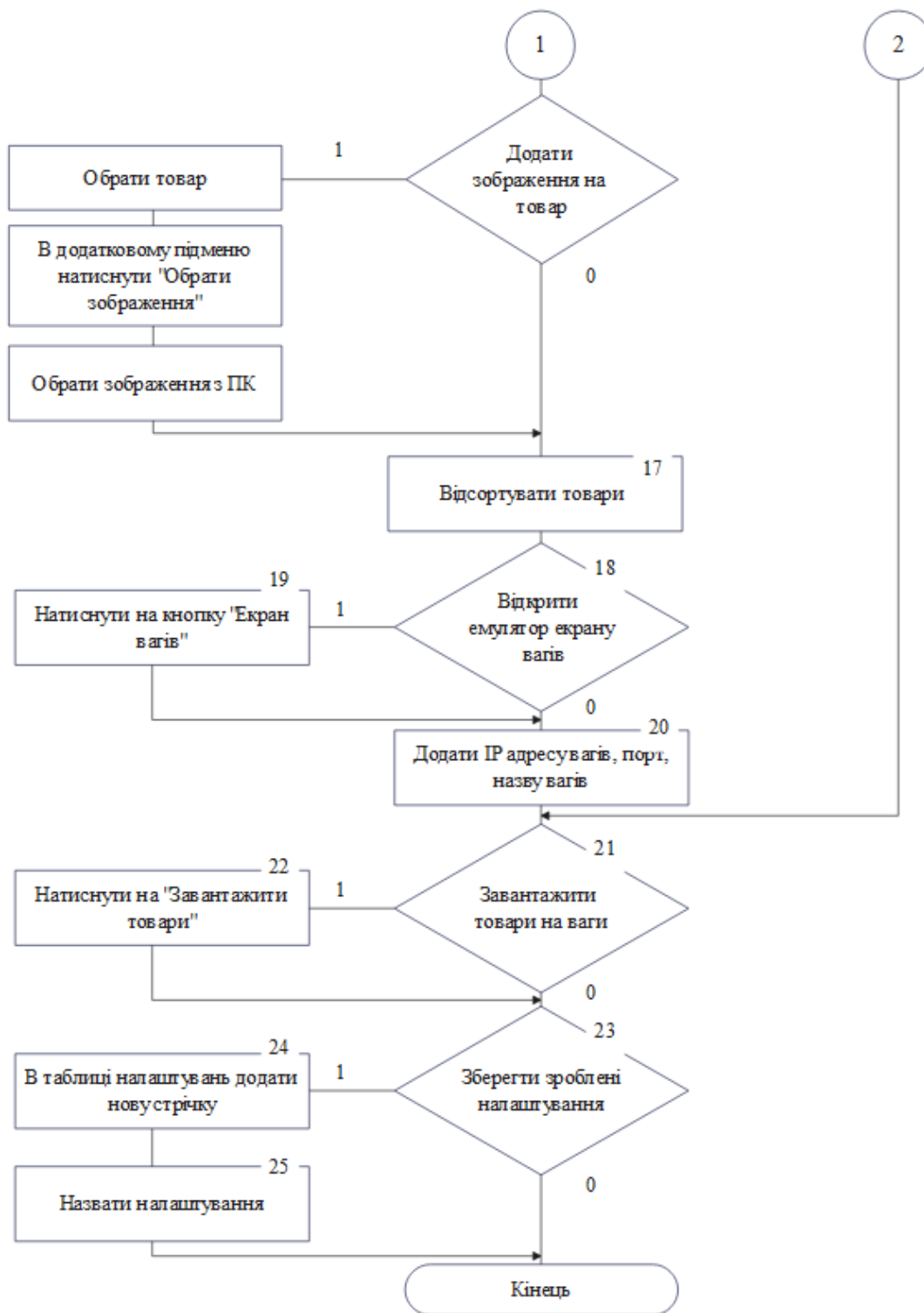


Рисунок В.5, аркуш 2

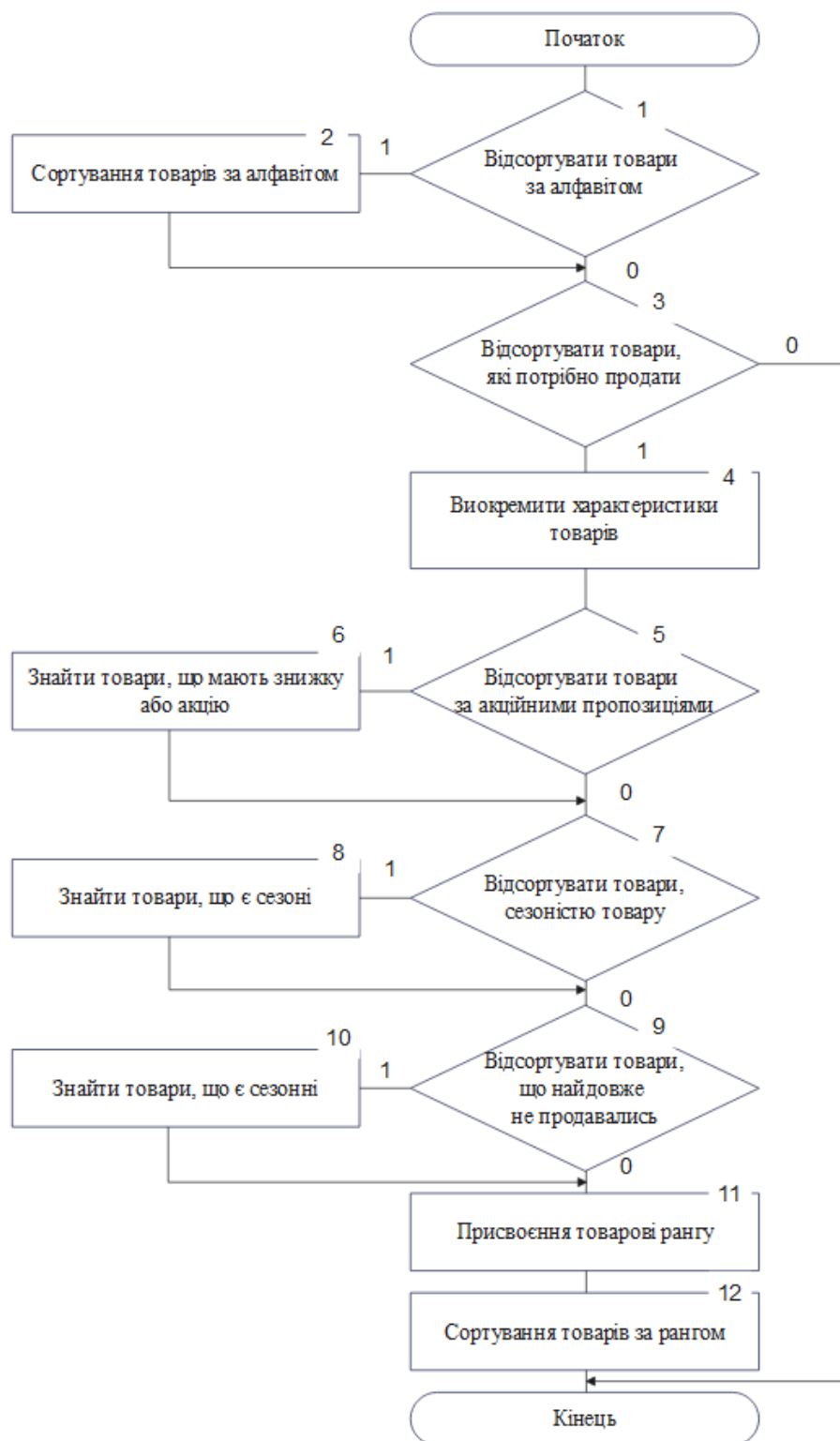


Рисунок В.6 – Схема алгоритму функціонування модуля сортування товарів перед завантаженням на сенсорні ваги самообслуговування



Рисунок В.7 – Діаграма створення товарів на емуляторі екрану вагів

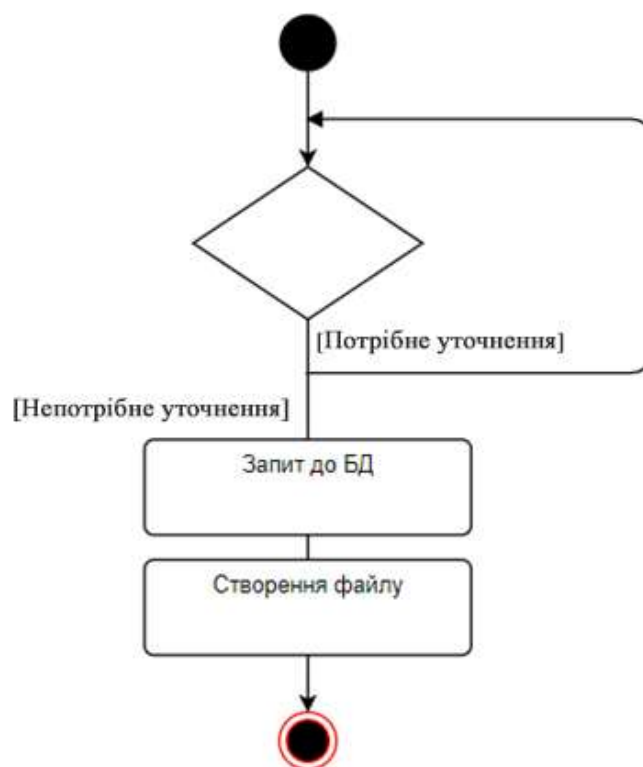


Рисунок В.8 – Діаграма створення файлу XML для завантаження інформації на ваги

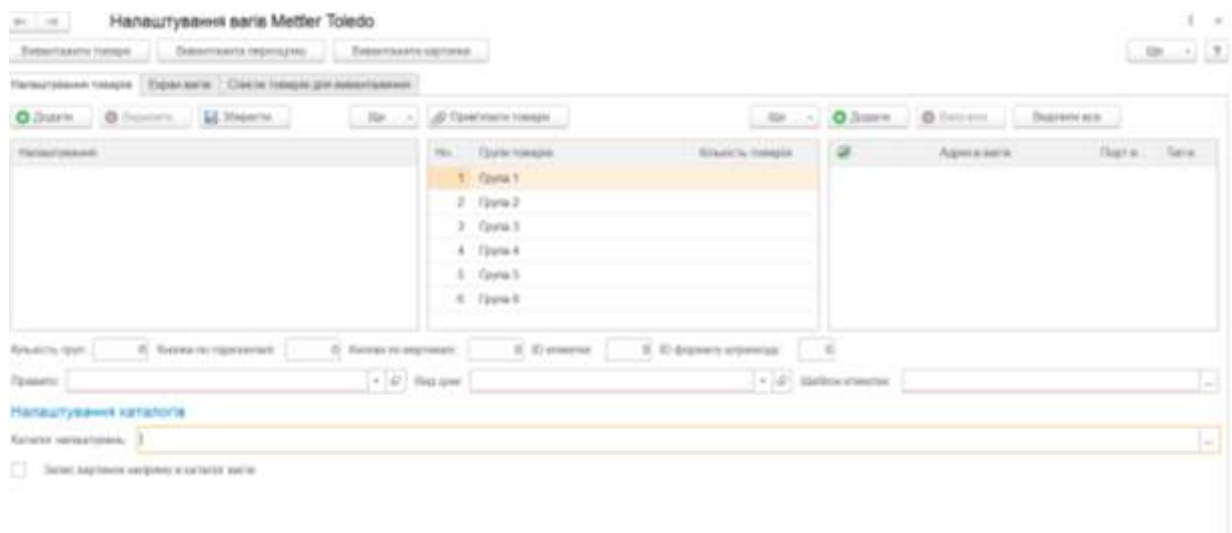


Рисунок В.9 – Загальний вигляд інтерфейсного вікна налаштувань для роботи з вагами



Рисунок В.10 – Загальний вигляд інтерфейсного вікна емулятора екрану вагів після додавання зображення на товари

Інструкція користувача

Відкривши програмний засіб перед користувачем з'являється головний екран, на якому розташовані налаштування. Для того, щоб створити категорії товарів, потрібно в полі «Кількість груп», ввести потрібну кількість (максимальна кількість 6) (рис. Г.1).

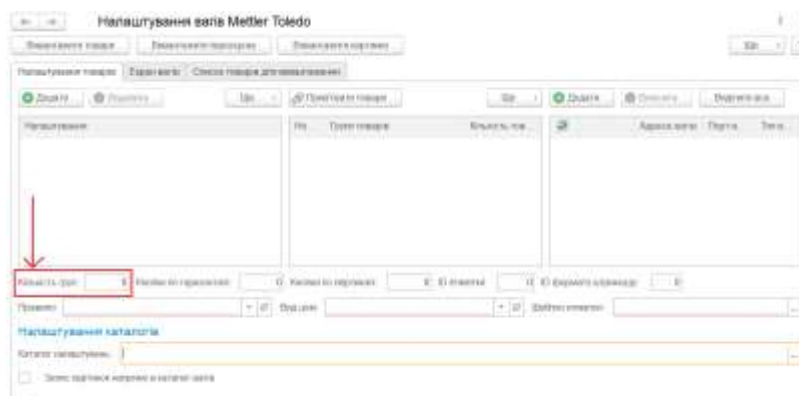


Рисунок Г. 1 – Загальний вигляд інтерфейсного вікна головних налаштувань

Після встановлення кількості груп товарів, в таблиці груп (рис. Г. 2), з'являться стандартні групи, які користувач може змінити на потрібні йому назви. Перша колонка такої таблиці показує номер групи, друга колонка – назва групи, третя колонка – кількість товарів в групі.

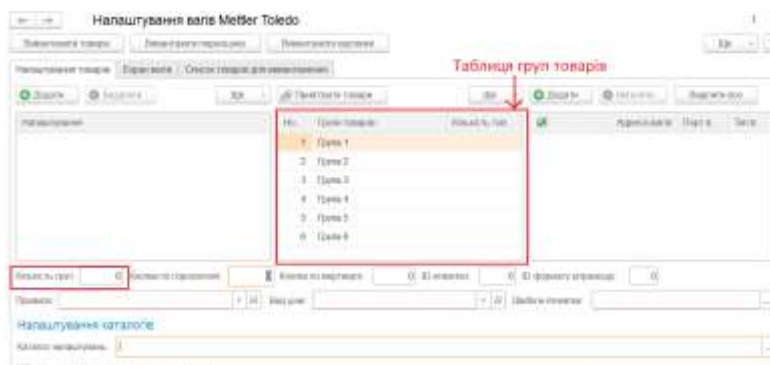


Рисунок Г. 2 – Загальний вигляд інтерфейсного вікна створення категорій товарів

Для того, щоб переглянути як виглядають створені групи на екрані вагів, потрібно перейти на сторінку з назвою «Екран вагів» (рис. Г. 3).

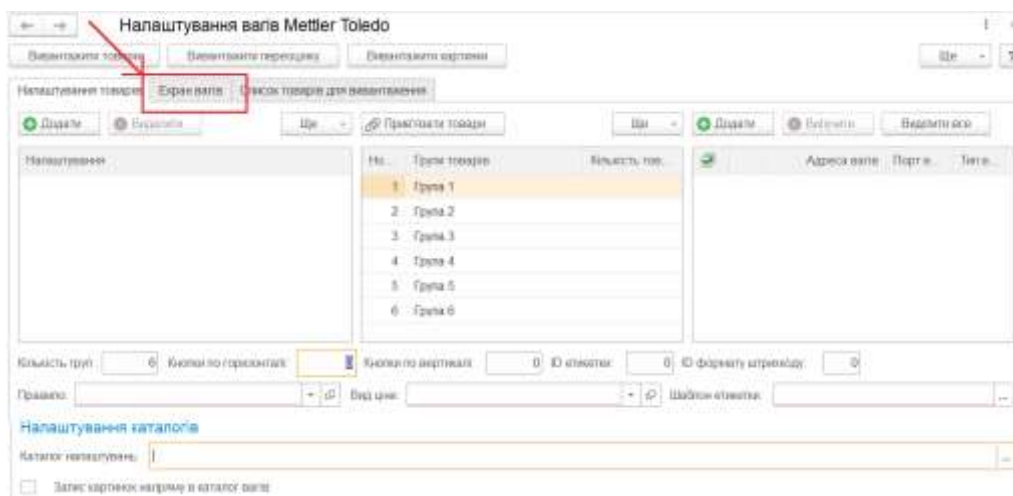


Рисунок Г. 3 – Загальний вигляд інтерфейсного вікна відкриття емулятора екрану вагів

Після чого відкриється екран вагів (рис. Г. 4).



Рисунок Г. 4 – Загальний вигляд інтерфейсного вікна емулятор екрану вагів

Для того, щоб додати товари до кожної групи потрібно задати такі параметри (рис. Г. 5):

1. Кількість товарів по горизонталі, що буде відображатись на вагах.
2. Кількість товарів по вертикалі.
3. Правило обміну (обрати з випадаючого списку, якщо список пустий створити нове правило обміну).
4. Вид цін (обрати з випадаючого списку).

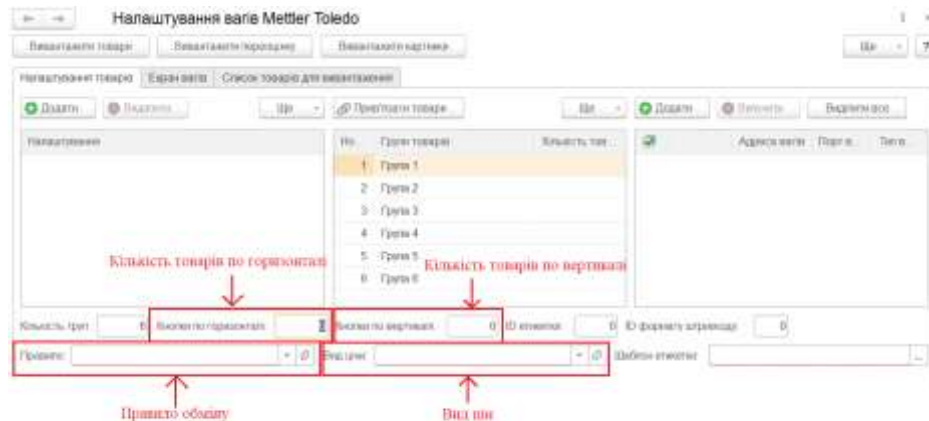


Рисунок Г. 5 – Загальний вигляд інтерфейсного вікна налаштування для вибору товарів

Задавши такі параметри, в користувача з'являється можливість додати товари до груп товарів. Зробити це можна натиснувши кнопку «Прив'язати товари» (рис. Г. 6). За замовчуванням група, в яку будуть додаватись товари – це поточний рядок таблиці груп.



Рисунок Г. 6 – Загальний вигляд інтерфейсного вікна з кнопкою для відкриття вікна з вибором товарів

Після чого відкриється вікно додавання товарів до груп товарів (рис. Г.7).

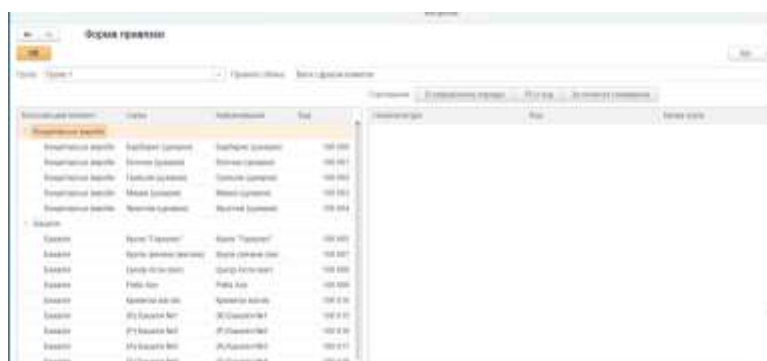


Рисунок Г.7 – Загальний вигляд інтерфейсного вікна для вибору товарів в категорію

В полі вибору групи, є випадаючий список створених раніше груп товарів. Зліва представлена таблиця всіх існуючих вагових товарів, а справа таблиця товарів, що будуть прив'язані до обраної групи. Прив'язати товари до обраної групи можна перенесенням їх з лівої таблиці в праву. Перенести можна як по одному товару так і всю групу. Список доданих товарів до групи можна відсортувати за PLU кодом товару, в алфавітному порядку та за попитом товарів серед споживачів, натиснувши кнопку «PLU», «В алфавітному порядку» та «За попитом в споживачів» відповідно.

Прив'язавши необхідні товари до групи потрібно натиснути «ОК». Для того, щоб переглянути таблицю всіх товарів, можна перейти на сторінку «Список товарів для завантаження» (рис. Г. 8).

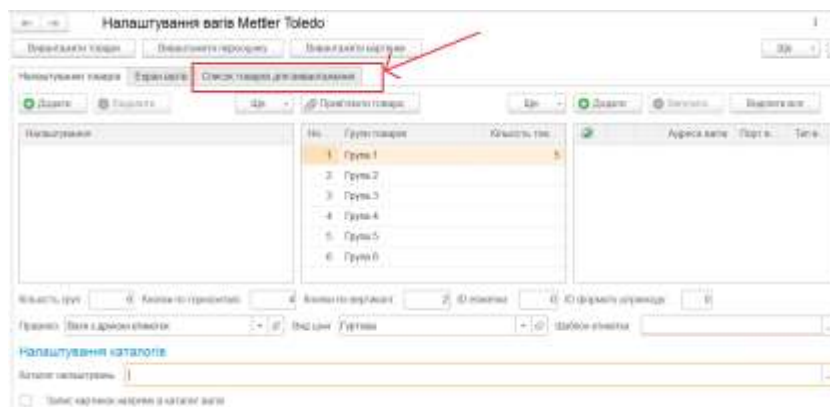


Рисунок Г. 8 – Загальний вигляд інтерфейсного вікна для відкриття вікна зі списком всіх товарів

На цій сторінці представлені всі обрані товари. В таблиці наведено назви товарів, назви груп до яких вони входять, їхнє розташування на вагах та файл зображення. Додати товар до будь-якої групи, можна з цього вікна також, натиснувши кнопку «Додати», відкриється вікно роботи з товарами (рис. Г. 9).

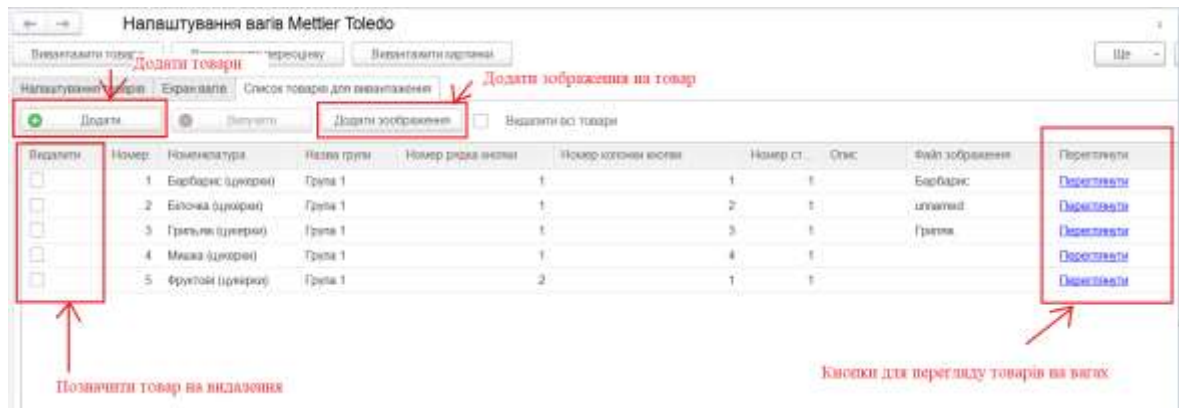


Рисунок Г. 9 – Загальний вигляд інтерфейсного вікна зі списком всіх товарів для завантаження

Для того, щоб додати зображення на товар, або змінити існуюче потрібно обрати рядок з потрібним товаром та натиснути кнопку «Додати зображення», або натиснути на потрібний товар правою кнопкою миші та у додатковому підменю обрати «Додати зображення».

З такої таблиці можна також видаляти товари, поставивши галочку біля товару, який ви хочете видалити та натиснути кнопку «Видалити».

Переглянути вигляд кнопок на вагах, користувач може за допомогою кнопки «Переглянути», що є біля кожного товару, або обравши сторінку «Екран вагів» (рис. Г.3), після чого відкриється (рис. Г.4). На рис. Г.4 зображені створені категорії товарів, натиснувши на будь-яку з них з'являться товари відповідної категорії (рис. Г.10).

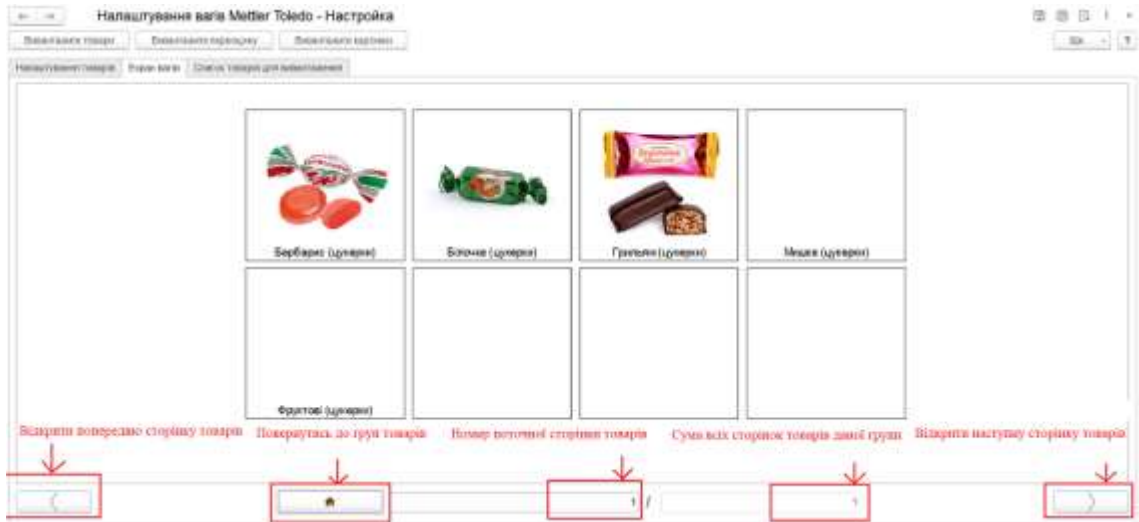


Рисунок Г.10 – Загальний вигляд інтерфейсного вікна з емулятором екрану вагів із зображеннями товарів обраної категорії

Для того, щоб додати ваги на які будуть завантажуватись товари, потрібно в таблиці зі списком вагів натиснути кнопку «Додати» (рис. Г.11). Після чого відкриється вікно для заповнення інформації про ваги. В такому вікні потрібно заповнити поля: IP адреса вагів, порт вагів та обрати тип вагів із випадваючого списку.

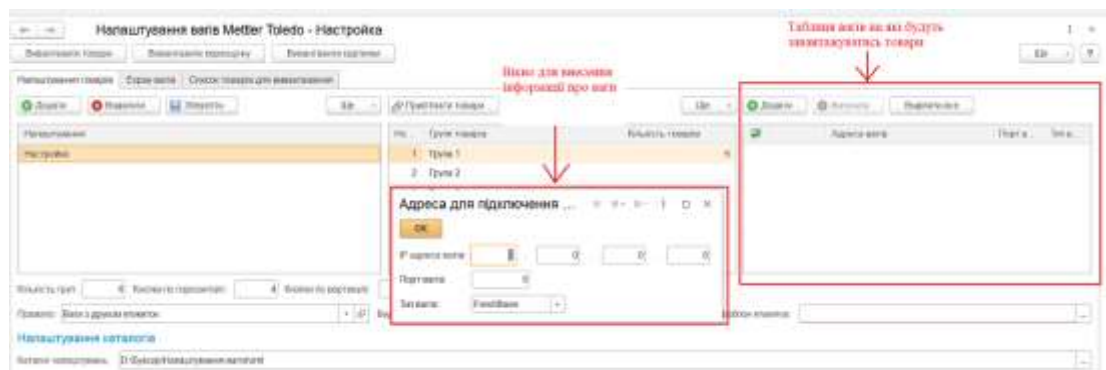


Рисунок Г.11 – Загальний вигляд інтерфейсного вікна для додавання вагів

Перед завантаженням товарів на ваги потрібно заповнити параметри зображені на рис. Г.12.

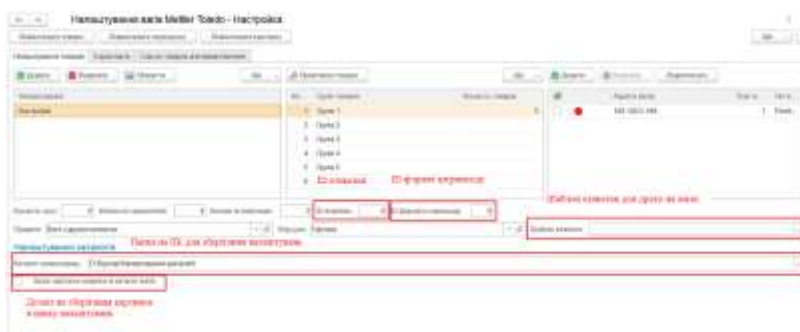


Рисунок Г.12 – Загальний вигляд інтерфейсного вікна з налаштуваннями для завантаження товарів на ваги та збереження зроблених налаштувань

Для того, щоб зберегти зроблені користувачем налаштування потрібно створити новий рядок в таблиці налаштувань (рис. Г.12), дати назву такому налаштуванню та натиснути кнопку «Зберегти». При наступному відкритті таке налаштування буде доступне, для того, щоб його відкрити достатньо обрати потрібний рядок таблиці налаштувань. Налаштування можна видалити, обравши потрібний рядок та натиснувши кнопку «Видалити».

Якщо користувач видалить рядок з вже існуючим налаштуванням та натисне зберегти, то налаштування, що було під цим іменем видалиться, а на його місце запишеться нове (рис. Г.13).

Якщо користувач випадково перезаписав налаштування, в нього є можливість повернутись до попереднього налаштування натиснувши кнопку «Відновити попередні налаштування» (рис. Г.14), видаливши поточне та повернувши на його місце попередньо видалене налаштування. Повернення можливе тільки на 1 крок назад.

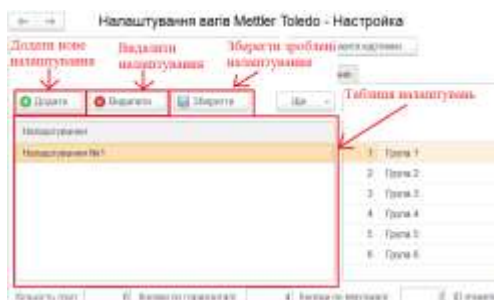


Рисунок Г.13 – Загальний вигляд інтерфейсного вікна з таблицею для збереження налаштувань

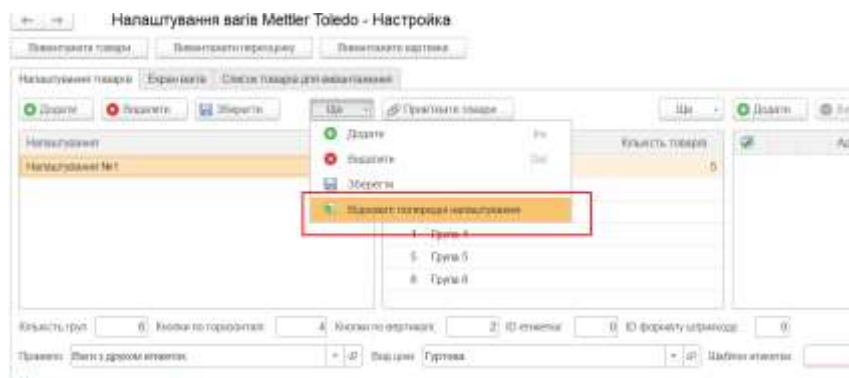


Рисунок Г.14 – Загальний вигляд інтерфейсного вікна з кнопкою для відновлення видалених налаштувань

При повторному відкритті в таблиці налаштувань будуть попередньо збережені налаштування. Якщо натиснути на рядок таблиці в якій потрібне налаштування, всі раніше заповнені параметри заповняться (рис. Г.15).

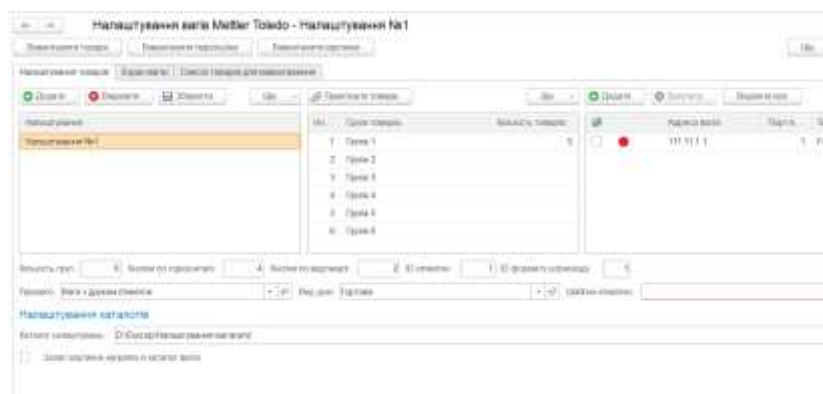


Рисунок Г.15 – Загальний вигляд інтерфейсного вікна заповнення всіх налаштувань із раніше збережених

Для завантаження всіх зроблених налаштувань потрібно натиснути на кнопку «Вивантажити товари» (рис. Г.16).

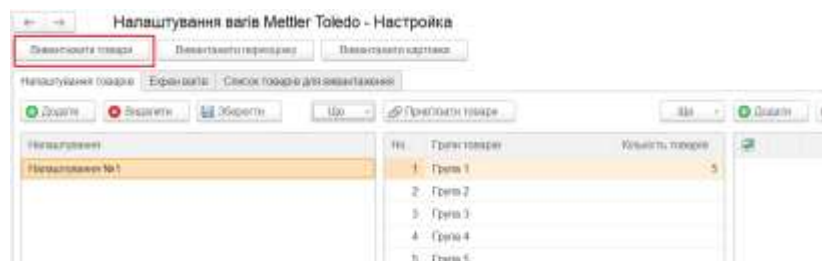


Рисунок Г.16 – Загальний вигляд інтерфейсного вікна з кнопкою для завантаження товарів на ваги

Для завантаження переоцінки потрібно натиснути кнопку «Вивантажити переоцінку» (рис. Г.17).

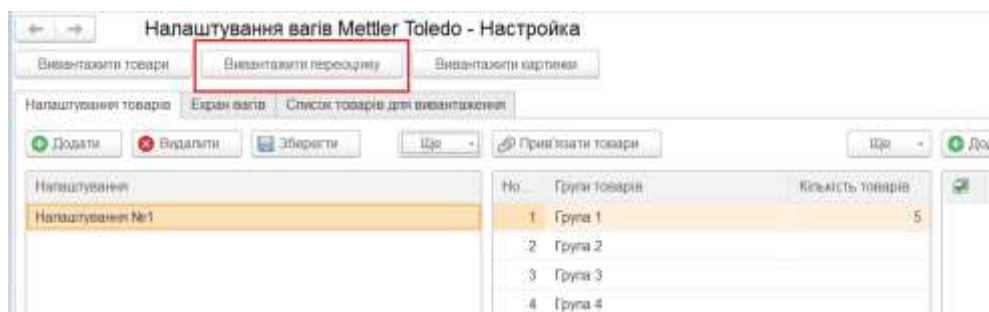


Рисунок Г.17 – Загальний вигляд інтерфейсного вікна з кнопкою для завантаження переоцінки на ваги

Для завантаження картинок потрібно натиснути на кнопку «Вивантажити товари» (рис. Г.18).

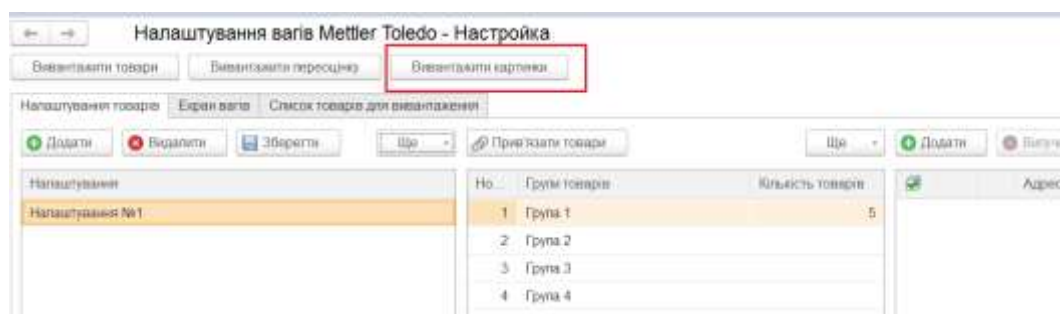


Рисунок Г.18 – Загальний вигляд інтерфейсного вікна з кнопкою для завантаження картинок товарів

Довідка про участь в конкурсі студентських наукових робіт

Д О В І Д К А

Надана БУКСИР Яні Олександрівні, ст. гр. ІКН-21м в тому, що вона брала участь у Всеукраїнському конкурсі студентських наукових робіт в 2021/2022 н.р.

За підсумками І-го туру наукова робота «Розробка програмного додатку для автоматизації роботи супермаркетів» була рекомендована для участі у II –му турі конкурсу і направлена до Центральноукраїнського НТУ у м. Кропивницький, який не проводився в зв'язку з введенням в країні воєнного стану (Наказ МОН України № 508 від 31 травня 2022).

Секретар оргкомітету  Людмила КОНДРАТЬЄВА




Підпис: *Л. Кондратьєвої*
ПОСВІДЧУЮ
Зак. канцелярією 

Довідка про впровадження розробки



Товариство з обмеженою відповідальністю "АКСІОМА В"
21021, м. Вінниця, вул. Космонавтів, буд. 30А, оф. 701,
тел. +38 (0432) 46-93-98, код за ЄДРПОУ 37490385,
ПІН 374903802289, № свід. 200014122, р/р 26009055311395,
банк ПАТ КБ "Приватбанк", МФО 302689

Вих. № 14 від "28" 11 2022

Довідка про впровадження

Довідка дана студентці Вінницького національного технічного університету групи ІКН-21м Буксир Яні Олександрівні в тому, що результати магістерської кваліфікаційної роботи «Інформаційна технологія автоматизації роботи супермаркетів» пройшли експериментальне випробування, відповідають замовленню, мають практичне значення і плануються до впровадження та подальшого використання.

Директор



В.М.Костинюк