

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра програмного забезпечення автоматизованих систем
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»

Завідувач кафедри

_____ проф. Приходько С.Б.

«___» _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «бакалавр»

**на тему: «Розробка програмного забезпечення для автоматизації управління
складом магазину електроніки «Tech Store»»**

Виконав: студент групи 3157ст

_____ 

(підпис, ПІБ)

Нагорний В. В.

Керівник роботи:

доцент каф. ПЗАС, к.т.н., доцент

(посада, науковий ступень вчене звання)

Макарова Л.М.

(підпис, ПІБ)

Миколаїв – 2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами

Кафедра програмного забезпечення автоматизованих систем

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

Гарант освітньої програми

_____ доц. Макарова Л.М.

(підпис)

«___» _____ 2022 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ на здобуття ступеня вищої освіти «бакалавр»

Студенту _____ Нагорному Владиславу Валерійовичу
(Прізвище, ім'я, по батькові)

1. Тема роботи: Розробка програмного забезпечення для автоматизації управління складом магазину електроніки «Tech Store»

Керівник роботи Макарова Лідія Миколаївна

Затверджені наказом ректора №256-уч від «11» травня 2022 р.

2. Термін подання роботи: 10.06.2022 р.

3. Вихідні дані по роботі: _____

4. Перелік питань, що належать до розробки (найменування розділів) _____
Титульний аркуш; Завдання на кваліфікаційну роботу; Анотація (українською та англійською мовами); Зміст; Перелік умовних позначень, символів, одиниць та термінів (при необхідності); Вступ; Опис предметної галузі та постановка задачі; Проєкт програмного забезпечення (ескізний, технічний та робочий); Результати розробки програмного забезпечення; Розділ з охорони праці; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма та методика випробувань програмного забезпечення).

5. Перелік презентаційних матеріалів

Тема кваліфікаційної роботи; Вступ; Аналіз роботи магазину; Аналіз готових рішень; Постанова задачі; Діаграма варіантів використання; Концептуальна модель; Діаграма діяльності; Фізична модель даних; Засоби розробки; Результат розробки; Висновок.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Гурець Н.В., ст. викладач		

7. Дата видачі завдання 11.05.2022 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу Вступ	24.03.2022	ПП*
2.	Опис та аналіз предметної галузі	28.03.2022	ПП*
3.	Постановка задачі	31.03.2022	ПП*
4.	Ескізний проєкт програмного забезпечення	27.04.2022	
5.	Технічний проєкт програмного забезпечення	06.05.2022	
6.	Робочий проєкт програмного забезпечення	13.05.2022	
7.	Підготовка розділу Результати розробки програмного забезпечення	17.05.2022	
8.	Підготовка розділу з охорони праці	18.05.2022	ПП*
9.	Підготовка розділу Висновки	19.05.2022	
10.	Оформлення списку використаних джерел та додатків	20.05.2022	
11.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версій роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	10.06.2022	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку)
12.	Підготовка презентації та доповіді	12.06.2022	
13.	Попередній захист роботи на засіданні кафедри ПЗАС	14.06.2022	
14.	Подання на кафедру ПЗАС електронних версій наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді	24.06.2022	

* - за результатами переддипломної практики (ПП), яка була з 21.03.2022 до 08.05.2022 р.

Студент

(підпис)

Нагорний В.В.

(ПІБ)

Керівник роботи

(підпис)

Макарова Л.М.

(ПІБ)

АНОТАЦІЯ

Кваліфікаційна робота присвячена розробці програмного забезпечення для автоматизації управління складом магазину електроніки «Tech Store».

Дана кваліфікаційна робота передбачає розв'язання практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме розробку програмного забезпечення для автоматизації управління складом магазину електроніки «Tech Store», котре дозволить працівникам магазину спростити та пришвидшити свою роботу в управлінні бізнес-процесами складської роботи підприємства.

У кваліфікаційній роботі міститься аналіз та опис предметної галузі за темою кваліфікаційної роботи «Розробка програмного забезпечення для автоматизації управління складом магазину електроніки «Tech Store»», а також постановка задачі та розділ з охорони праці, ескізний, технічний та робочий проекти ПЗ. Звіт містить чотири розділи та п'ять додатків – технічне завдання на розробку програмного забезпечення, текст програми, опис програми, інструкція користувача, програма та методика випробувань ПЗ.

Кваліфікаційна робота викладена на 99 сторінках друкованого тексту та містить 20 рисунків, 24 таблиці, список використаних джерел з 11 найменувань та 5 додатків.

ABSTRACT

Qualification work is devoted to the development of software for automation of warehouse management of the electronics store "Tech Store".

This qualification work involves solving the practical problem of software engineering, characterized by complex and uncertain conditions, namely the development of software to automate the management of the electronics store "Tech Store", which will allow store employees to simplify and speed up their work in business process management warehousing of the enterprise.

Qualification work contains an analysis and description of the subject area on the topic of qualification work "Development of software for automation of warehouse management of the electronics store "Tech Store", as well as problem statement and section on labor protection, sketch, technical and working software projects. The report contains four sections and five appendices - terms of reference for software development, program text, program description, user manual, program and software test methodology.

Qualification work is presented on 99 pages of printed text and contains 20 figures, 24 tables, a list of references from 11 names and 5 appendices.

ЗМІСТ

ВСТУП	7
1 АНАЛІЗ РОБОТИ МАГАЗИНУ ЕЛЕКТРОНІКИ «TECH STORE» ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ УПРАВЛІННЯ СКЛАДОМ МАГАЗИНУ	9
1.1 Опис та аналіз системи управління складом магазину електроніки «Tech Store»	9
1.2 Аналіз існуючих аналогів програмного забезпечення	12
1.3 Постановка задачі до розробки програмного забезпечення для автоматизації управління складом магазину електроніки «Tech Store»	16
2 ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ УПРАВЛІННЯ СКЛАДОМ МАГАЗИНУ ЕЛЕКТРОНІКИ «TECH STORE»	19
2.1 Ескізний проект програмного забезпечення	19
2.1.1 Контекстна діаграма	19
2.1.2 Діаграма варіантів використання	20
2.1.3 Специфікації варіантів використання	21
2.1.4 Концептуальна модель	26
2.1.5 Розробка ескізу інтерфейсу	27
2.2 Технічний проект програмного забезпечення	29
2.2.1 Логічна модель бази даних	29
2.2.2 Діаграма діяльності	30
2.2.3 Діаграма послідовності	31
2.2.4 Діаграма класів	32
2.2.5 Діаграма розгортання	33

2.3	Робочий проект програмного забезпечення.....	34
2.3.1	Вибір і обґрунтування інструментів розробки та СКБД.....	34
2.3.2	Фізична модель.....	36
2.3.3	Реалізація та тестування основних класів еквівалентності програмного забезпечення	39
2.3.4	Випробування програмного забезпечення	41
3	РЕЗУЛЬТАТИ РОЗРОБКИ ПЗ ДЛЯ АВТОМАТИЗАЦІЇ УПРАВЛІННЯ СКЛАДОМ МАГАЗИНУ ЕЛЕКТРОНІКИ «TECH STORE»	44
4	ОХОРОНА ПРАЦІ.....	46
4.1	Вступ	46
4.2	Основні законодавчі і нормативні документи, які регламентують охорону праці в Україні.....	47
4.3	Загальні вимоги до виробничих приміщень ЕОМ.....	48
4.4	Аналіз шкідливих та небезпечних факторів на робочому місці	51
4.5	Заходи щодо зменшення впливу шкідливих факторів.....	52
	ВИСНОВКИ.....	54
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	55
	ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ.....	57
	ДОДАТОК Б – ТЕКСТ ПРОГРАМИ	64
	ДОДАТОК В – ОПИС ПРОГРАМИ	85
	ДОДАТОК Г – ІНСТРУКЦІЯ КОРИСТУВАЧА	90
	ДОДАТОК Д – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	96

ВСТУП

На сьогоднішній день в усьому світі існує велика кількість магазинів електроніки, в яких здійснюється продаж різноманітних електронних товарів, а також їх зберігання та облік. Працівники таких магазинів у своїй роботі взаємодіють з великою кількістю товарів, що зберігаються на складі та на полицях магазину. Задля полегшення роботи працівників в управлінні складом окремого магазину, а також пришвидшення та спрощення процесу отримання точної інформації про конкретний товар та його місцезнаходження на складі, розробимо програмне забезпечення (ПЗ) для автоматизації управління складом магазину електроніки «Tech Store».

Кваліфікаційна робота передбачає розв'язання практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме розробку програмного забезпечення для автоматизації управління складом магазину електроніки «Tech Store», котре дозволить працівникам магазину спростити та пришвидшити свою роботу в управлінні бізнес-процесами складської роботи підприємства.

Автоматизація складських процесів – підвищує ефективність управління та конкурентоспроможність будь-якого бізнесу. Якщо інвентаризація, облік товарів, зберігання та багато інших процесів не автоматизовані, то оптимізації складського господарства бути не може. В цілому, якщо усі вищезгадані процеси будуть керуватися комп'ютером, то буде звільнено багато часу для вирішення інших завдань по складу. Імплементация WMS (з англ. Warehouse Management System) сприяє організації складських процесів підприємства та підвищує продуктивність [1].

Метою роботи є розробка програмного забезпечення, що буде забезпечувати комплексну автоматизацію управління складської роботи магазину електроніки «Tech Store». ПЗ буде розроблено з використанням бази

даних, тому частиною роботи буде завдання змодельовати базу даних, що відтворюватиме структуру складу, яка відображує дані про товари та пов'язану з ними інформацію.

Виділимо наступні цілі, виконання яких необхідне для успішної розробки ПЗ:

- Обстежити та проаналізувати предметну галузь, з метою виявлення усіх вимог до розроблюваного ПЗ.
- Розробити специфікацію модулів ПЗ.
- Розробити технічне завдання ПЗ.
- Розробити концептуальну модель ПЗ з метою виявлення відношень між сутностями бази даних.
- Виходячи з вимог до ПЗ, обрати найбільш вдалу комбінацію системи управління базами даних (СУБД) та програмних засобів.
- Розробити логічну та фізичну моделі ПЗ.
- Виконати кодування ПЗ відповідно до розроблених моделей та з використанням обраних програмних засобів.
- Виконати тестування та випробування ПЗ.

1 АНАЛІЗ РОБОТИ МАГАЗИНУ ЕЛЕКТРОНІКИ «TECH STORE» ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ УПРАВЛІННЯ СКЛАДОМ МАГАЗИНУ

1.1 Опис та аналіз системи управління складом магазину електроніки «Tech Store»

WMS – система управління складом, яка забезпечує комплексну автоматизацію та оптимізацію всіх складських бізнес-процесів. Дана система є операційною програмою управління складським господарством: логістикою, обліком запасів, плануванням і т.д. Завдяки цій системі можна зменшити витрати і збільшити прозорість складських операцій. Всі бізнес-процеси, що виконуються на складі (облік товарів, їх зберігання та переміщення, збір замовлень, підготовка комплектів до відправки, інвентаризація) при недостатньому рівні автоматизації втрачають керованість. Чим масштабніше складське господарство і ширша номенклатура товарів, тим менш ефективно ручне управління, та вище ймовірність виникнення помилок, затримок і втрат.

Класифікація WMS-систем [1]:

- Системи початкового рівня (склади невеликих компаній, магазинів з невеликою номенклатурою).
- Коробкові системи управління складом (склади площею 1000-10 000 м² з великою номенклатурою, але невисоким товарообігом).
- Системи, що адаптуються (великі логістичні компанії, розподільні центри, склади площею більше 5000 м²).
- Конфігуровані системи (склади площею понад 5000 м² з великою номенклатурою і високим товарообігом).

Впровадження системи управління складом має економічний сенс всюди, де здійснюється зберігання, облік і переміщення будь-яких одиниць зберігання:

товарів, поштових відправлень, архівних даних та ін. Як показує практика, застосування WMS має сенс не тільки в великих логістичних центрах, а й у порівняно невеликих складських господарствах, розподільних центрах, архівах, на виробництві тощо. Незалежно від масштабу сховища, автоматизація призводить до значного зменшення витрат часу і коштів.

Функціонал WMS-системи керування складом:

- Приймання товару:

Система WMS управляє роботою складу за попередньо налаштованими бізнес-процесами, виконання яких є обов'язковою умовою при виконанні всіх операцій, у тому числі при прийманні товару. Процес приймання товару може бути налаштований під вимоги користувачів системи, але основне завдання WMS-системи забезпечити приймання товару з мінімальними помилками, заощаджуючи при цьому час при виконанні операції.

- Облік товару:

WMS система для складу веде облік товару, надає користувачам актуальну інформацію щодо його кількості, дозволяє мінімізувати затарювання складу непотрібними залишками. Економія складського простору та безперебійна робота складу досягається за рахунок оптимального розподілу та зберігання запасів.

- Оптимізація процесу зберігання:

Система WMS надає користувачам можливість моделювання ефективних схем зберігання різних товарів, що враховують їх характеристики, наприклад, такі як, вага товару або його попит (швидкість оборотності товарних запасів). Це дозволяє організувати процес зберігання таким чином, що більш затребувані або важкі товари розташовуватимуться ближче до зони відвантаження або товар, що відвантажуються разом, зберігатиметься поруч один з одним, за рахунок чого часу для його обробки потрібно менше. Врахування численних факторів зберігання забезпечує ефективну роботу складу.

- Управління персоналом:

Централізоване управління складом у вигляді складської системи WMS скорочується необхідність у великій кількості персоналу. Оптимізація робочого фонду стає можливою, зокрема, за допомогою скорочення частоти інвентаризації товару. WMS програма дозволяє проводити інвентаризацію товару без втручання у повсякденну роботу складу. Скорочення витрат на оплату праці дозволяє знизити поточні (операційні) витрати на утримання складу та підвищити ефективність роботи всього підприємства. Вимірювання ключових показників ефективності роботи складу (KPI складу) підвищує ефективність роботи, дозволяє вимірювати показники ефективності, перевіряти виконання та результативність роботи, формувати форму звітності, налаштовувати систему мотивації та норми оплати праці.

- Документообіг:

WMS-система управління складом здатна автоматизувати всі процеси, усуваючи необхідність ведення паперового документообігу, що потребує значних ресурсів. Користувачі можуть мати загальний доступ до бази даних, що забезпечує працівників необхідною інформацією для швидкої та якісної роботи.

- Комплектація та відвантаження:

Програма управління складом забезпечує якісне складання замовлень, це означає, що процес складання буде виконаний за складськими стандартами, методами FIFO, FEFO, FPFO та LIFO. WMS забезпечить, щоб правильно зібране замовлення було доставлено на правильну адресу в потрібний час.

- Обслуговування клієнтів:

WMS-система управління складом підвищує якість обслуговування клієнтів за рахунок швидкої та безпомилкової обробки замовлень поклажедавців та своєчасної доставки. Висока якість обслуговування підвищує конкурентоспроможність компанії, дозволяє формувати лояльність поточних замовників та залучати нових клієнтів.

- Управління складом та контроль:

Для підприємств, яким потрібні розширені можливості контролю, програма WMS пропонує відстежувати товар за різними характеристиками: серійні номери, терміни придатності, товарні коди тощо. Питання щодо повернення та умов гарантії якості товару швидко вирішуються шляхом можливості відстеження каналів поставок.

Впровадження системи управління складом має економічний сенс всюди, де здійснюється зберігання, облік і переміщення будь-яких одиниць зберігання: товарів, поштових відправлень, архівних даних та ін. Як показує практика, застосування WMS має сенс не тільки в великих логістичних центрах, а й у порівняно невеликих складських господарствах, розподільних центрах, архівах, на виробництві тощо. Незалежно від масштабу сховища, автоматизація призводить до значного зменшення витрат часу і коштів.

Магазин електроніки, для якого розробляється ПЗ – «Tech Store» – це маленький магазин, розташований в спальному районі Миколаєва. У даному магазині здійснюється продаж техніки, а саме: комп'ютерів та їх комплектуючих, смартфонів, навушників, ноутбуків, музикальних пристроїв, аксесуарів для телефонів тощо. І як у будь-якому магазині, там також є склад, в якому зберігаються товари на продаж. Система управління складом допоможе організувати раціональне розміщення і зберігання продукції магазину, прискорить і спростить отримання інформації про кількість і розташування товару, оптимізує використання складських площ та спростить виконання усіх функцій для персоналу.

1.2 Аналіз існуючих аналогів програмного забезпечення

На сьогоднішній день існує багато аналогів програмного забезпечення для автоматизації роботи складу магазинів. Розглянемо та проаналізуємо декілька існуючих аналогічних ПЗ та розглянемо їх переваги і недоліки,

порівняно з розробляючимим у даній роботі ПЗ, аби зробити висновок про доцільність розробки ПЗ для нашого магазину електроніки «Tech Store».

«1С:Підприємство» – це програмний продукт, який призначений для автоматизації процесів обліку (бухгалтерський, кадровий, управлінський і т.п.) і управління виробничими процесами на підприємстві. За допомогою даної програми, завдяки її універсальності і багатofункціональності, можна автоматизувати бізнес-процеси використовуючи її типові і галузеві рішення.

По своїй суті це – програмна оболонка, яка систематизує базу даних і дає можливість здійснювати до неї доступ та виконувати супутні операції за виразно короткий термін [2]. На її базі побудовано досить багато додаткових прикладних рішень (Бухгалтерія, Управління торгівлею, Зарплата і Кадри, Управління торговим підприємством, Торгівля і Склад, Комплексний облік, а також вузькоспрямована 1С: Бухгалтерія для України.

До переваг «1С:Підприємство» відносять:

- Конфігурація є відкритою – за допомогою відкритого програмного коду існує можливість адаптування будь-якого компоненту 1С для власних потреб окремо взятого підприємства.

- Індивідуальне проектування – кожен користувач може створювати й згодом доопрацьовувати проекти, що враховують специфіку бізнес-процесів тієї чи іншої організації.

- Технологічна платформа – вона єдина для всіх програмних продуктів, і це означає, що всі прикладні рішення можуть бути досить швидко стандартизовані за сучасними вимогами, а усі проекти – масштабуються за потребою.

- Простота продукту – незважаючи на широкий функціонал, 1С не вважається специфічним програмним забезпеченням, і тому навчитися роботи в ньому може практично кожен співробітник.

- Універсальність – продукт підходить і може бути адаптивний абсолютно для всіх сфер бізнесу.

- Недоліками ж вважаються такі особливості програми:
- Платна адаптація базового продукту – створення тих чи інших конфігурацій вимагає залучення стороннього фахівця, який згодом буде підтримувати, обслуговувати та допрацьовувати компоненту.
- Основні конфігурації – Управління торгівлею, Підприємство, Бухгалтерія без адаптації не підходять для сфери послуг, спеціальних об'єктів або роздрібної торгівлі – знову ж таки, необхідне доопрацювання фаховим спеціалістом.
- Оновлення не є безкоштовним. Усі більш новітні версії програми потрібно купувати.
- Багато функцій є зайвими для нашої задачі.

Інтерфейс «1С:Підприємство» представлений на рисунку 1.1.

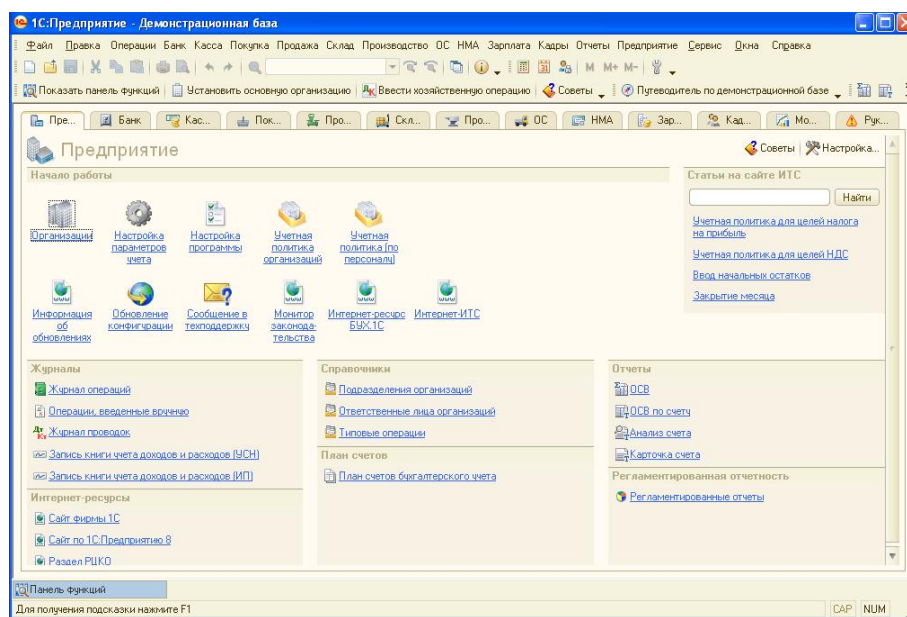


Рисунок 1.1 – Інтерфейс «1С:Підприємство» [2]

WMS UIS – існуюча система управління складом, яка автоматизує та оптимізує всі складські процеси на підприємстві від прийому товару до відвантаження [3]. Система управління складом WMS UIS забезпечує точність

відвантажень до 99,5%. Працює з такими компаніями, як: Nestle, Нова пошта, Симпатик, Стітус, Вітмарк (Jaffa).

Задачі, які вирішує дана WMS:

- Рациональне використання площ складу. Завдяки застосуванню WMS вдається експлуатувати кожен квадратний метр приміщення.

- Оптимальне складування продукції. Система нагадує, як потрібно розкласти товари на складі. Наприклад, ближче до вантажної зони розміщується важка продукція, що швидко псується або продукція з підвищеним попитом. Система може враховувати різні фактори, покращуючи продуктивність.

- Інформація про розташування товарів на складі. Не потрібно витрачати багато часу на пошук певної продукції. Комп'ютер підкаже, де є той чи інший товар.

- Інвентаризація без зупинки роботи складу. WMS UIS дозволяє проводити інвентаризацію складу частинами, без зупинки його роботи.

- KPI складу та співробітників. Залишки та KPI співробітників доступні в реальному часі (кількість відвантажених/прийнятих товарів, швидкість роботи кожного співробітника, залишки).

- Спрощення документообігу. Система складського обліку дозволяє «розвантажити» співробітників, прибравши потребу багато працювати з документами. Час, що звільнився, фахівці можуть витрачати на дійсно важливі завдання.

Недоліками такої системи є висока ціна та складний функціонал, орієнтація на великі магазини.

Інтерфейс WMS UIS представлений на рисунку 1.2.

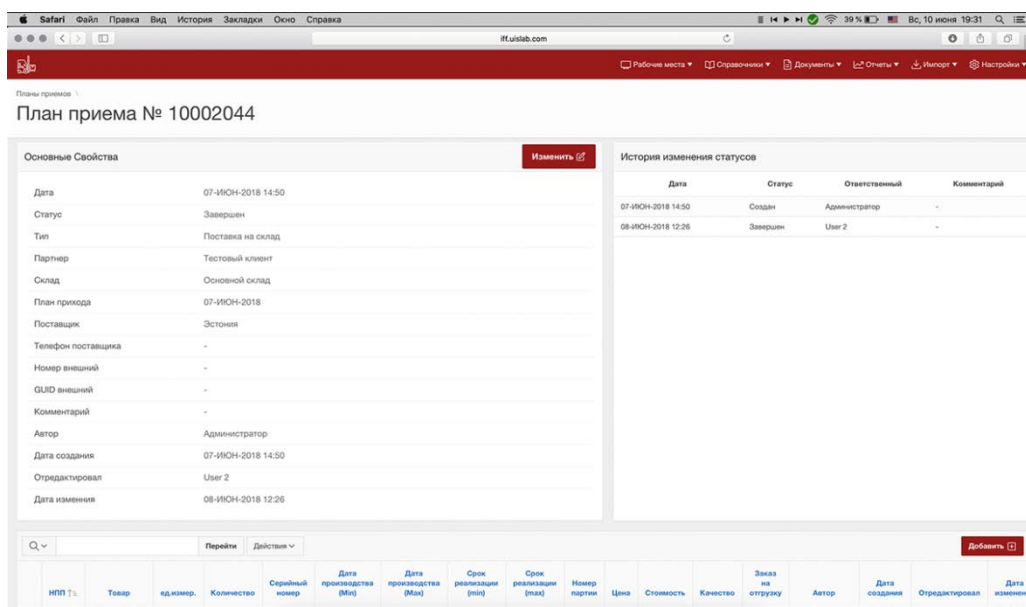


Рисунок 1.2 – Интерфейс WMS UIS [3]

Отже, проаналізувавши аналогічні існуючі програмні рішення для автоматизації управління складом, прийшли до висновку, що система, яка б підходила для нашого маленького магазину з продажу електроніки, яка була б простою у використанні, мала невеликий функціонал та була безкоштовною, не було виявлено.

1.3 Постановка задачі до розробки програмного забезпечення для автоматизації управління складом магазину електроніки «Tech Store»

В результаті аналізу предметної галузі та аналізу аналогічних існуючих програмних продуктів, була сформульована наступна задача – розробити програмне забезпечення для автоматизації управління складом магазину електронної техніки «Tech Store». ПЗ буде розроблене у вигляді додатку. Метою розробки ПЗ є спрощене виконання всіх функцій складської роботи, усунення малоефективної роботи з паперами, прискорити отримання інформації про товари на складі, їх кількість та розташування, зменшити

навантаження на персонал та полегшити їх роботу. А це, в свою чергу, підвищить продуктивність роботи закладу, і як наслідок, підвищить його прибуток.

ПЗ у вигляді додатку буде мати функцію авторизації та розподілятися на адміністратора та працівника складу.

Для адміністратора мають бути реалізовані наступні функції:

- додавати нового користувача в систему;
- редагувати користувача;
- видаляти користувача;
- створювати нове замовлення товарів;
- додавати новий товар на склад;
- редагувати дані про товар;
- видаляти товар;
- переглядати усі товари на складі;
- додавати нового постачальника;
- редагувати постачальника;
- видаляти постачальника;
- додавати нову категорію товарів;
- редагувати категорію;
- видаляти категорію товарів.

Для працівника складу мають бути реалізовані наступні функції:

- створювати нове замовлення товарів;
- додавати новий товар на склад;
- редагувати дані про товар;
- видаляти товар;
- переглядати усі товари на складі;
- додавати нового постачальника;
- редагувати постачальника;
- видаляти постачальника;

- додавати нову категорію товарів;
- редагувати категорію;
- видаляти категорію товарів.

Вхідні дані:

- інформація про категорії товарів;
- інформація про товари;
- інформація про користувачів системи;
- інформація про замовлення;
- інформація про постачальників.

Вихідні дані:

- дані про категорії товарів;
- дані про товари;
- дані про користувачів системи;
- дані про замовлення;
- дані про постачальників.

2 ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ УПРАВЛІННЯ СКЛАДОМ МАГАЗИНУ ЕЛЕКТРОНІКИ «TECH STORE»

2.1 Ескізний проект програмного забезпечення

2.1.1 Контекстна діаграма

Проектування будь-якого програмного забезпечення починається з розробки ескізного проекту, у якому представляються результати зовнішнього проектування програмного забезпечення.

При проектуванні даного програмного забезпечення було обрано об'єктно-орієнтований підхід та, як наслідок, мову моделювання UML.

UML може бути застосовано на всіх етапах розробки прикладних програм. Різні види діаграм, які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису програмних систем [4].

Першою при проектуванні програмної системи розробляється контекстна діаграма, що представляє собою загальний опис системи та її взаємодію з зовнішнім середовищем.

Контекстна діаграма для програмного забезпечення для автоматизації управління складом представлена на рисунку 2.1.

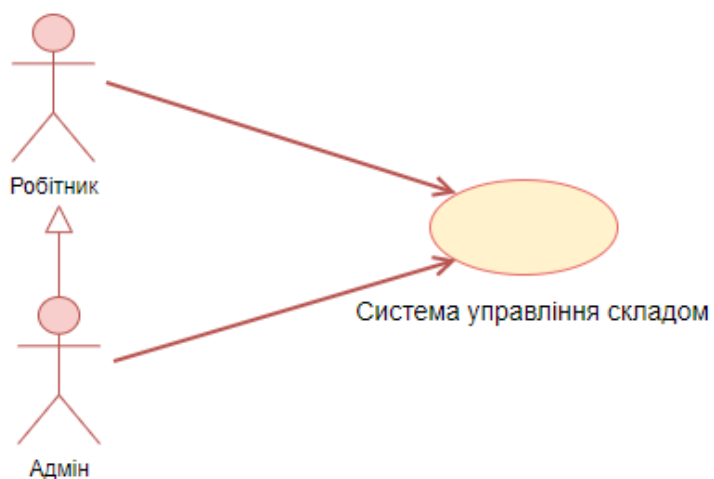


Рисунок 2.1 – Контекстна діаграма системи управління складом

На контекстній діаграмі показаний загальний опис взаємодії системи з користувачами. Для більш детального опису потрібно побудувати діаграму варіантів використання.

2.1.2 Діаграма варіантів використання

Для уточнення функціональності, виявлення схеми функціональної структури, визначення модулів, які входять до її складу, і вивчення динаміки роботи ПЗ в цілому побудуємо діаграму варіантів використання, яка відобразить системні прецеденти, системне оточення та зв'язки між ними.

Діаграма варіантів використання автоматизації управління складом представлена на рисунку 2.2.



Рисунок 2.2 – Діаграма варіантів використання для системи управління складом

Після того, як була побудована діаграма варіантів використання, розпишемо специфікації для кожного з варіантів.

2.1.3 Специфікації варіантів використання

Розглянемо більш детально кожний варіант використання надаючи опис потоків подій. Специфікації варіантів використання можна переглянути в таблицях 2.1-2.14.

Таблиця 2.1 – Специфікація варіанту використання «Додавання товару»

Характеристика	Опис
1	2
Призначення	призначений для додавання товару на склад
Учасники	активний суб'єкт «Робітник» та «Адміністратор»
Передумови	Авторизація
Основний потік подій	завантажується вікно списку товарів, куди вносяться необхідні дані
Альтернативний потік подій	Відсутній
Опис	варіант використання дозволяє додати товар до складу

Таблиця 2.2 – Специфікація варіанту використання «Редагування товару»

Характеристика	Опис
1	2
Призначення	призначений для редагування товару на складі
Учасники	активний суб'єкт «Робітник» та «Адміністратор»
Передумови	Авторизація
Основний потік подій	завантажується вікно списку товарів, де можна обрати певний товар і редагувати його дані
Альтернативний потік подій	Авторизація
Опис	варіант використання дозволяє редагувати дані товару

Таблиця 2.3 – Специфікація варіанту використання «Видалення товару»

Характеристика	Опис
1	2
Призначення	призначений для видалення товару зі складу
Учасники	активний суб'єкт «Робітник» та «Адміністратор»
Передумови	Авторизація
Основний потік подій	завантажується вікно списку товарів, де можна обрати певний товар і видалити його
Альтернативний потік подій	Авторизація
Опис	варіант використання дозволяє видалити товар зі складу

Таблиця 2.4 – Специфікація варіанту використання «Створення заказу поставки»

Характеристика	Опис
1	2
Призначення	призначений для створення заказу поставки на склад
Учасники	активний суб'єкт «Робітник» та «Адміністратор»
Передумови	Авторизація
Основний потік подій	завантажується вікно заказу, де можна створити новий заказ
Альтернативний потік подій	Авторизація
Опис	варіант використання дозволяє створювати новий заказ

Таблиця 2.5 – Специфікація варіанту використання «Додавання категорії»

Характеристика	Опис
1	2
Призначення	призначений для додавання категорії товару
Учасники	активний суб'єкт «Робітник» та «Адміністратор»
Передумови	Авторизація
Основний потік подій	завантажується вікно списку категорій, де можна додати нову категорію товару
Альтернативний потік подій	Авторизація
Опис	варіант використання дозволяє додати категорію товару

Таблиця 2.6 – Специфікація варіанту використання «Редагування категорії»

Характеристика	Опис
1	2
Призначення	призначений для редагування категорії товару
Учасники	активний суб'єкт «Робітник» та «Адміністратор»
Передумови	Авторизація
Основний потік подій	завантажується вікно списку категорій, де можна редагувати категорію
Альтернативний потік подій	Авторизація
Опис	варіант використання дозволяє редагувати категорію товару

Таблиця 2.7 – Специфікація варіанту використання «Видалення категорії»

Характеристика	Опис
1	2
Призначення	призначений для видалення категорії товару
Учасники	активний суб'єкт «Робітник» та «Адміністратор»
Передумови	Авторизація
Основний потік подій	завантажується вікно списку категорій, де можна видалити категорію
Альтернативний потік подій	Авторизація
Опис	варіант використання дозволяє видалити категорію товару

Таблиця 2.8 – Специфікація варіанту використання «Додавання постачальника»

Характеристика	Опис
1	2
Призначення	призначений для додавання постачальника
Учасники	активний суб'єкт «Робітник» та «Адміністратор»
Передумови	Авторизація
Основний потік подій	завантажується вікно постачальників, де можна додати нового
Альтернативний потік подій	Авторизація
Опис	варіант використання дозволяє додати постачальника

Таблиця 2.9 – Специфікація варіанту використання «Редагування постачальника»

Характеристика	Опис
1	2
Призначення	призначений для редагування постачальника
Учасники	активний суб'єкт «Робітник» та «Адміністратор»
Передумови	Авторизація
Основний потік подій	завантажується вікно списку постачальників, де можна обрати певного постачальника і редагувати його дані
Альтернативний потік подій	Авторизація
Опис	варіант використання дозволяє редагувати постачальника

Таблиця 2.10 – Специфікація варіанту використання «Видалення постачальника»

Характеристика	Опис
1	2
Призначення	призначений для видалення постачальника
Учасники	активний суб'єкт «Робітник» та «Адміністратор»
Передумови	Авторизація
Основний потік подій	завантажується вікно списку товарів, де можна обрати певний товар і редагувати його дані
Альтернативний потік подій	Авторизація
Опис	варіант використання дозволяє видалити постачальника

Таблиця 2.11 – Специфікація варіанту використання «Додавання користувача»

Характеристика	Опис
1	2
Призначення	призначений для додавання нового користувача додатку
Учасники	активний суб'єкт «Адміністратор»
Передумови	Авторизація
Основний потік подій	завантажується вікно користувачів, де можна додати нового
Альтернативний потік подій	Авторизація
Опис	варіант використання дозволяє додати нового користувача

Таблиця 2.12 – Специфікація варіанту використання «Редагування користувача»

Характеристика	Опис
1	2
Призначення	призначений для редагування користувача додатку
Учасники	активний суб'єкт «Адміністратор»
Передумови	Авторизація
Основний потік подій	завантажується вікно користувачів, де можна редагувати обраного користувача
Альтернативний потік подій	Авторизація
Опис	варіант використання дозволяє редагувати користувача

Таблиця 2.13 – Специфікація варіанту використання «Видалення користувача»

Характеристика	Опис
1	2
Призначення	призначений для видалення користувача додатку
Учасники	активний суб'єкт «Адміністратор»
Передумови	Авторизація
Основний потік подій	завантажується вікно користувачів, де можна видалити користувача
Альтернативний потік подій	Авторизація
Опис	варіант використання дозволяє видалити користувача

Таблиця 2.14 – Специфікація варіанту використання «Авторизація»

Характеристика	Опис
1	2
Призначення	призначений для авторизації користувача у додатку
Учасники	активний суб'єкт «Робітник» та «Адміністратор»
Передумови	Авторизація
Основний потік подій	завантажується вікно входу у додаток, де потрібно ввести логін та пароль
Альтернативний потік подій	Авторизація
Опис	варіант використання дозволяє увійти у додаток

Після детального розглядання специфікацій варіантів використання, побудуємо модель даних.

2.1.4 Концептуальна модель

Концептуальна модель даних – це модель даних на найбільш абстрактному рівні або модель даних на рівні зведення. Інформація, специфічна для платформи, та інша інформація про реалізацію, наприклад визначення інтерфейсу чи процедури, вилучаються з цієї моделі даних [5]. Концептуальна модель представлена на рисунку 2.3 і представлена за допомогою ER-діаграми.

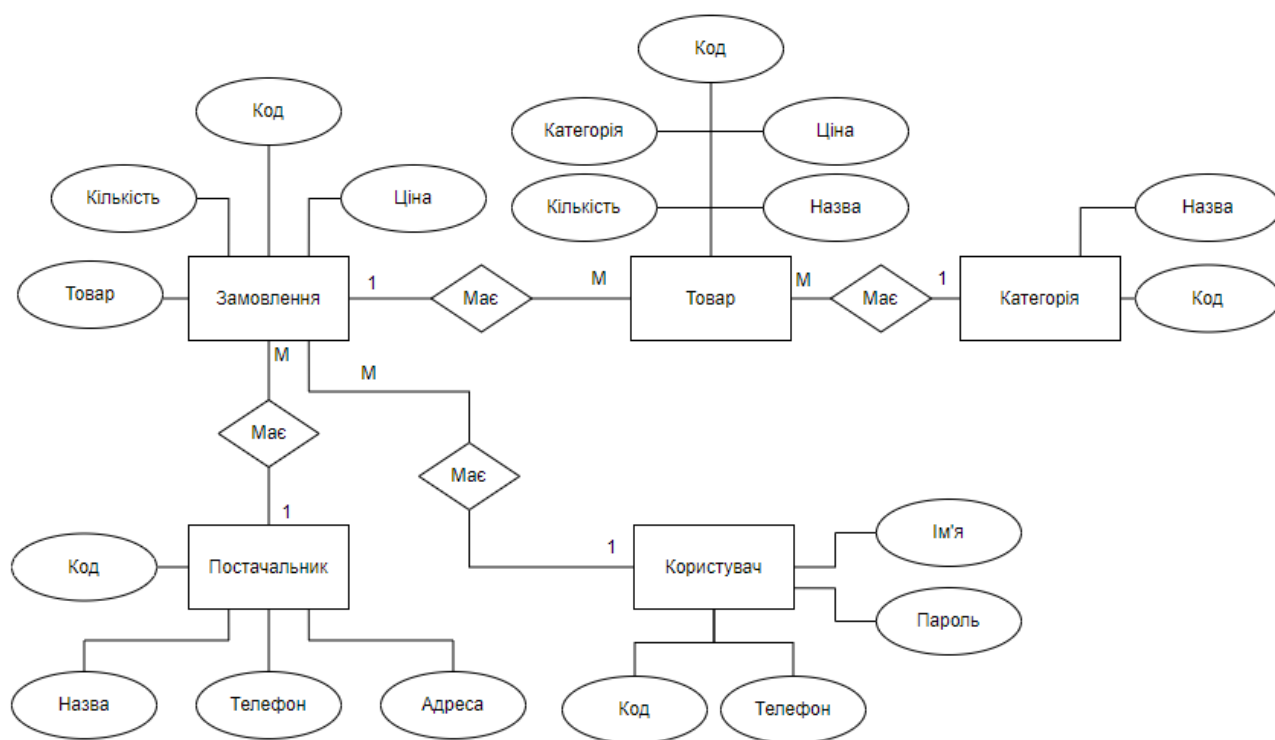


Рисунок 2.3 – Концептуальна модель для системи управління складом

Таблиця 2.15 – Опис зв'язків концептуальної моделі

Сутності	Тип зв'язку	Опис
1	2	3
Замовлення – Товар	1:M	В одному замовленні товарів може бути багато
Категорія – Товар	1:M	В одній категорії може бути багато товарів
Постачальник – Замовлення	1:M	Один постачальник може виконувати багато замовлень
Користувач – Замовлення	1:M	Один користувач робить багато замовлень

Після побудови концептуальної моделі, наступною моделлю даних, яка буде відображати дані програми, буде логічна модель даних.

2.1.5 Розробка ескізу інтерфейсу

Важливим етапом при проектуванні програмного забезпечення є розробка інтерфейсу.

Інтерфейс додатку повинен максимально інформативно і в стислому об'ємі відображати необхідну користувачеві інформацію про додаток. На головній сторінці необхідно помістити основне меню додатку (для навігації по його структурі), а також повинно бути вікно аутентифікації (входу зареєстрованих користувачів).

Важливою частиною проектування є створення користувацького інтерфейсу, основними формами завжди були реєстрація та сторінка логіну. Після запуску програми відкриватиметься форма авторизації, після якої буде завантажуватися головне вікно додатку з головним меню. Вікно з списком товарів та його редагування представлена на рисунку 2.4.

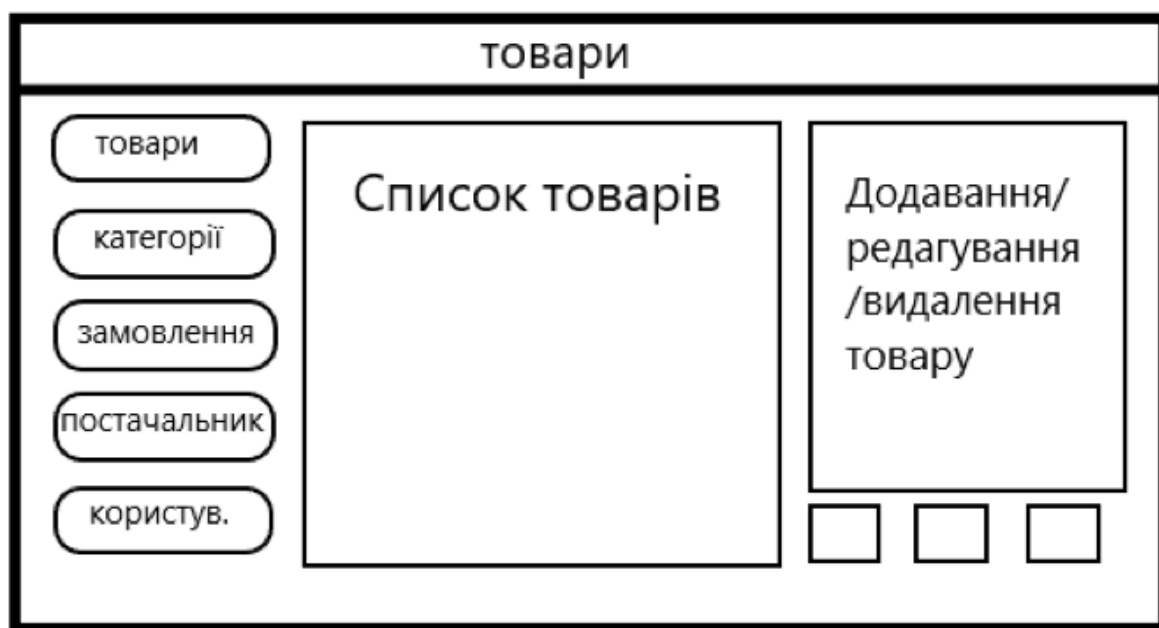


Рисунок 2.4 – Макет вікна товарів

На цьому рисунку ми бачимо ескіз вікна зі списком товарів, де зліва є кнопки – посилання на інші вікна додатку, в центрі сам список товарів, а праворуч – поля для введення даних про товар.

2.2 Технічний проект програмного забезпечення

2.2.1 Логічна модель бази даних

Логічна модель — модель даних конкретної предметної області, виражена незалежно від конкретного продукту керування базами даних або технології зберігання (фізична модель даних), але в термінах структур даних, таких як реляційні таблиці та колонки, об'єктно-орієнтовані класи чи теги XML. Вона є протилежністю концептуальній моделі даних, яка описує семантику організації без посилання на технологію [6]. Логічна модель показана на рисунку 2.5.

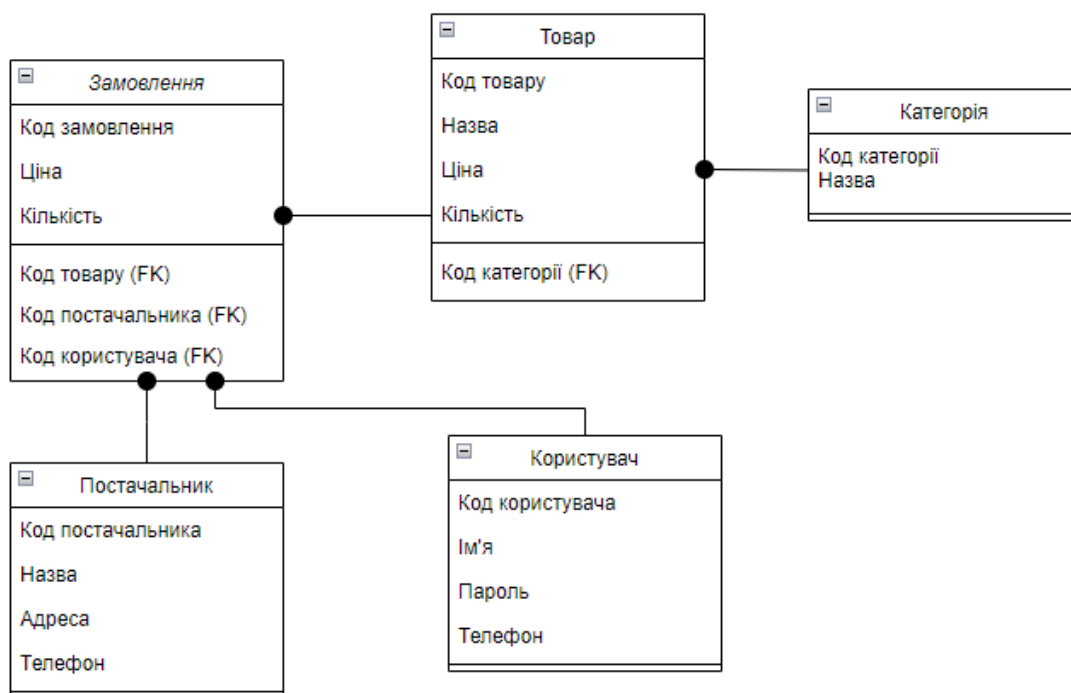


Рисунок 2.5 – Логічна модель для системи управління складом

Наступною моделю баз даних буде фізична модель, яка буде створена безпосередньо у середовищі керування базами даних.

2.2.2 Діаграма діяльності

Діаграма діяльності – в UML, візуальне представлення графу діяльності. Граф діяльності є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій.

Дія є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів. Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності.

Діаграма діяльності при виконанні варіанта використання «Додавання товару» представлена на рисунку 2.6.



Рисунок 2.6 – Діаграма діяльності для варіанту використання «Додавання товару»

На цій діаграмі показано, що для того, щоб додати товар, потрібно вибрати вікно «Товари» та ввести потрібні дані, а саме вписати усі поля та вибрати категорію, бо усі поля повинні бути заповнені, а потім натиснути кнопку додати. Але для цього всього потрібно спочатку увійти у систему та зареєструватися.

2.2.3 Діаграма послідовності

Діаграма послідовності – відображає взаємодії об'єктів впорядкованих за часом. Зокрема, діаграма відображає задіяні об'єкти та послідовність відправлених повідомлень.

Діаграма послідовності для всієї системи показана на рисунку 2.6.

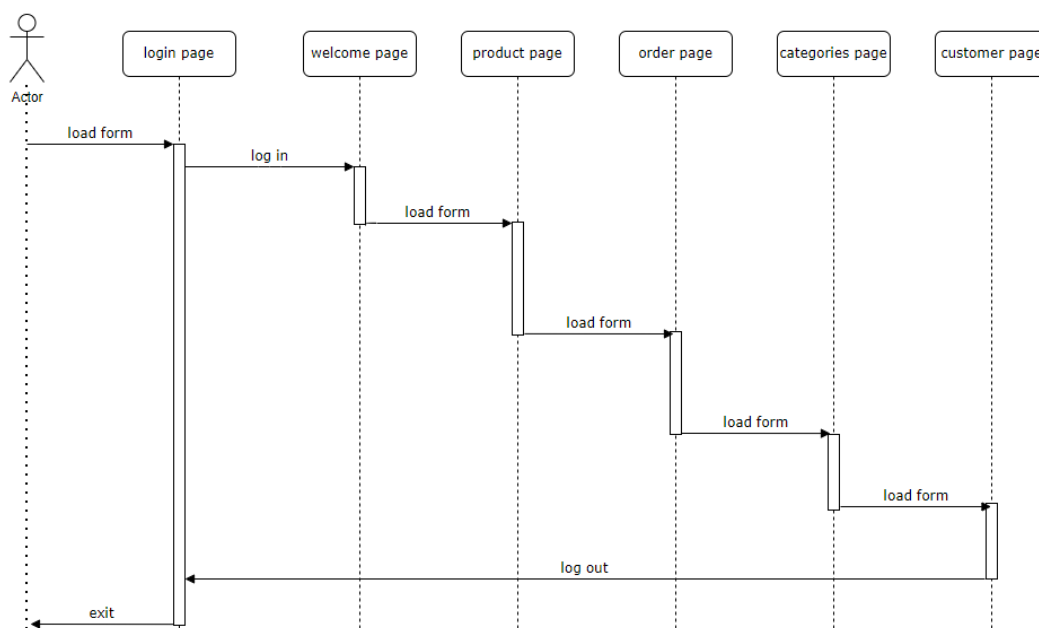


Рисунок 2.6 – Діаграма послідовності для користувача системи управління складом

Після того, як була побудована діаграма послідовності, побудуємо діаграму класів, де детально розпишемо усі класи у програмі.

2.2.4 Діаграма класів

Діаграма класів — статичне представлення структури моделі. Відображає статичні елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів, також, може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм.

Діаграма класів служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. На цій

діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

Діаграма класів представлена на рисунку 2.7.

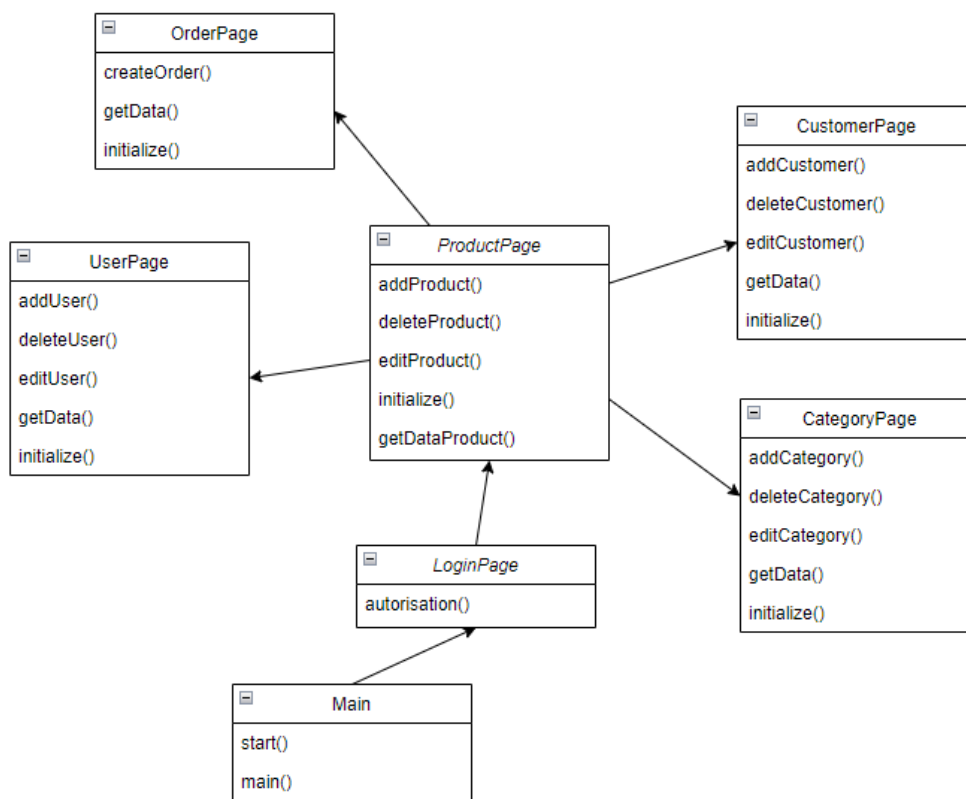


Рисунок 2.7 – Діаграма класів

На діаграмі класів показані класи-контролери, котрі містять функціонал вікон додатку.

2.2.5 Діаграма розгортання

Діаграма розгортання — діаграма в UML, на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах. Компоненти відповідають представленню робочих екземплярів одиниць коду. Компоненти, що не мають представлення під час роботи програми на таких діаграмах не відображаються; натомість, їх можна

відобразити на діаграмах компонент. Діаграма розгортання відображає робочі екземпляри компонент, а діаграма компонент, натомість, відображає зв'язки між типами компонент.

Діаграма розгортання представлена на рисунку 2.8.

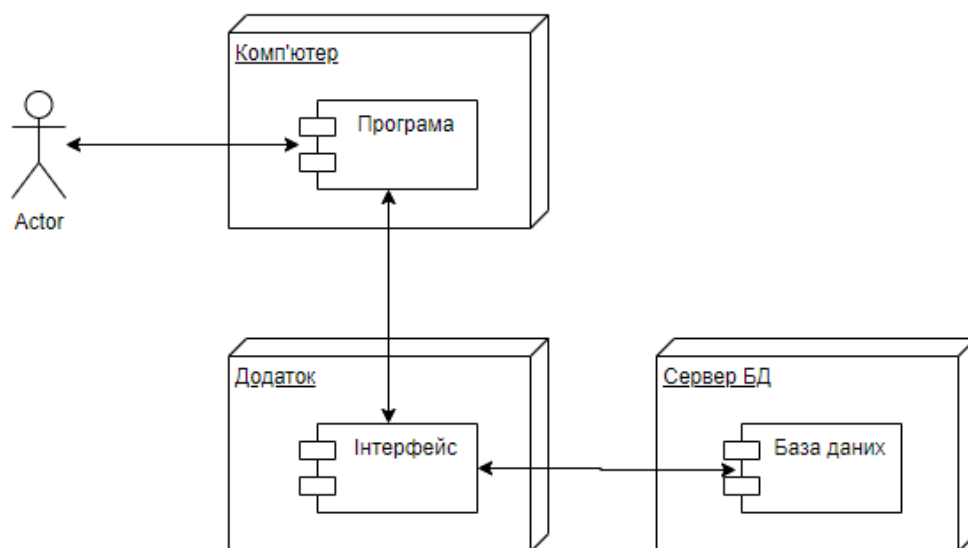


Рисунок 2.8 – Діаграма розгортання

Після побудови діаграми розгортання, наступним кроком буде робочий проект ПЗ.

2.3 Робочий проект програмного забезпечення

2.3.1 Вибір і обґрунтування інструментів розробки та СКБД

Для створення ПЗ «система для автоматизації управління складом магазину електроніки «Tech Store»» були обрані такі засоби для розробки:

- мова програмування Java;
- середовище розробки JetBrains IntelliJ IDEA;

- платформа на основі Java для створення програм з насиченим графічним інтерфейсом JavaFX та програма Scene Builder;
- СКБД MySQL Workbench.

Java є суворо типізованою об'єктно-орієнтованою мовою програмування, за допомогою якої можна створювати різні прикладні додатки для комп'ютерів, смартфонів, планшетів та інших інтелектуальних пристроїв. Але в основному, Java використовується для створення серверних додатків і мобільного ПЗ. Також – це основа прикладних додатків для системи Android. Особливістю програм на Java є те, що вони можуть запускатись на будь-яких комп'ютеризованих пристроях, які працюють під різними операційними системами, причому без повторної компіляції коду.

ПЗ JetBrains IntelliJ IDEA – це одне з найпопулярніших середовищ для розробки мовою Java. IntelliJ IDEA представляє собою комплекс інтегрованих інструментів програмування, до якого входять: автоматизований редактор вихідних текстів, передові інструменти для рефакторингу коду, вбудовану підтримку технологій J2EE, механізми інтеграції з середовищем тестування Ant/JUnit та системами керування версіями, унікальний інструмент оптим Code Inspection та інноваційний візуальний конструктор графічних інтерфейсів.

Унікальні можливості JetBrains IntelliJ IDEA позбавляють програміста вантажу рутинної роботи, допомагають своєчасно усунути помилки і підвищити якість коду, піднімаючи продуктивність розробника на нову висоту.

JavaFX — платформа на основі Java для створення програм з насиченим графічним інтерфейсом. Може використовуватися як для створення настільних програм, що запускаються безпосередньо з-під операційних систем, так і для інтернет-застосунків (RIA), що працюють у браузерях, і для програм на мобільних пристроях. JavaFX покликана замінити бібліотеку Swing, яка використовувалася раніше. Платформа JavaFX конкурує з Microsoft Silverlight, Adobe Flash та аналогічними системами.

MySQL – це система управління реляційними базами даних [7]. У реляційній базі даних дані зберігаються в окремих таблицях, завдяки чому досягається вииграш у швидкості й гнучкості. Таблиці зв'язуються між собою за допомогою відносин, завдяки чому забезпечується можливість поєднувати при виконанні запиту дані з декількох таблиць. SQL як частина системи MySQL можна охарактеризувати як мову структурованих запитів, що використовується для доступу до баз даних. ПЗ з відкритим кодом. Застосовувати його і модифікувати може будь-хто. Таке ПЗ можна отримувати за допомогою Internet і використовувати безкоштовно. При цьому кожен користувач може вивчити вихідний код і змінити його у відповідності зі своїми потребами.

MySQL Workbench – це інструмент для візуального проектування та моделювання бази даних для реляційної бази даних сервера MySQL [8]. Це полегшує створення нових фізичних моделей даних та модифікацію існуючих баз даних MySQL із зворотним/прямим інжинірингом та функціями управління змінами. Мета MySQL Workbench – надати інтерфейс для більш простої та структурованої роботи з базами даних.

2.3.2 Фізична модель

Фізична модель даних (або проектування бази даних) — подання дизайну даних як реалізованого чи призначеного для реалізації у системі керування базами даних. У життєвому циклі проекту вона типово походить від логічної моделі даних, хоча вона може бути зворотньо розроблена з даної реалізації бази даних. Завершена фізична модель даних включатиме всі артефакти бази даних, необхідні для створення відношень між таблицями чи для досягнення мети продуктивності, як-от індексів, визначень обмежень, зв'язаних і секціонованих таблиць або кластерів.

У даній кваліфікаційній роботі розроблено наступні таблиці бази даних:

- таблиця «Товар»;

- таблиця «Категорія»;
- таблиця «Користувач»;
- таблиця «Замовлення»;
- таблиця «Постачальник».

Структуру таблиці «Товар» представлено у таблиці 2.16.

Таблиця 2.16 – Структура таблиці «Product»

Ідентифікатор поля	Ознака ключа	Тип даних	Обмеження	Опис
idproduct	PK	INT	NOT NULL	Код товару
name		VARCHAR	NOT NULL	Назва товару
price		VARCHAR	NOT NULL	Ціна
quantity		INT	NOT NULL	Кількість
idCategoryfk	FK	INT	NOT NULL	Код категорії

Структуру таблиці «Категорія» представлено у таблиці 2.17.

Таблиця 2.17 – Структура таблиці «Category»

Ідентифікатор поля	Ознака ключа	Тип даних	Обмеження	Опис
idcategory	PK	INT	NOT NULL	Код категорії
name		VARCHAR	NOT NULL	Назва категорії

Структуру таблиці «Користувач» представлено у таблиці 2.18.

Таблиця 2.18 – Структура таблиці «User»

Ідентифікатор поля	Ознака ключа	Тип даних	Обмеження	Опис
iduser	PK	INT	NOT NULL	Код користувача
username		VARCHAR	NOT NULL	Ім'я
password		VARCHAR	NOT NULL	Пароль
phone		VARCHAR	NOT NULL	Номер телефону

Структуру таблиці «Замовлення» представлено у таблиці 2.19.

Таблиця 2.19 – Структура таблиці «Order»

Ідентифікатор поля	Ознака ключа	Тип даних	Обмеження	Опис
idorder	PK	INT	NOT NULL	Код користувача
cost		VARCHAR	NOT NULL	Ім'я
quantity		VARCHAR	NOT NULL	Пароль
product_fk		INT	NOT NULL	Номер телефону
customer_fk		INT	NOT NULL	
user_fk		INT	NOT NULL	

Структуру таблиці «Замовник» представлено у таблиці 2.20.

Таблиця 2.20 – Структура таблиці «Customer»

Ідентифікатор поля	Ознака ключа	Тип даних	Обмеження	Опис
idcustomer	PK	INT	NOT NULL	Код постачальника
name		VARCHAR	NOT NULL	Ім'я
adress		VARCHAR	NOT NULL	Адреса
phone		VARCHAR	NOT NULL	Номер телефону

Фізична модель бази даних для системи управління складом показана на рисунку 2.9.

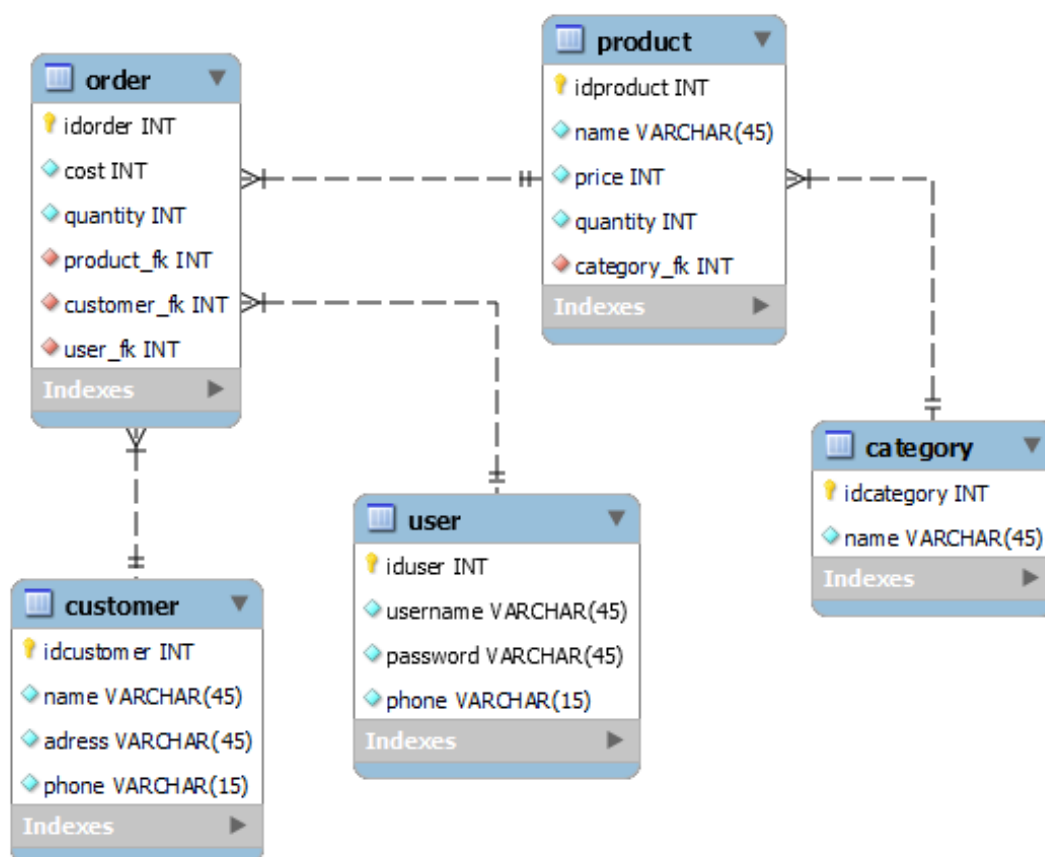


Рисунок 2.9 – Фізична модель БД для системи управління складом

Фізична модель, що представлена вище, показує таблиці, їх поля, унікальні та зовнішні ключі та зв'язки між ними, безпосередньо у системі керування базами даних MySQLWorkbench.

2.3.3 Реалізація та тестування основних класів еквівалентності програмного забезпечення

У даному підрозділі представлено тестування варіанту використання «Авторизація». Для тестування варіанту використання «Авторизація» був використаний метод еквівалентної розбивки. Виділені класи еквівалентності, представлені в табл. 2.21.

Таблиця 2.21 – Класи еквівалентності для тестування варіанту використання «Авторизація»

Вхідні дані	Правильні класи	Неправильні класи
Логін	Логін введено вірно (1)	Введений логін не існує(2), Логін не був введений (3)
Логін	Пароль введено вірно (4)	Введений пароль не існує (5), Пароль не був введений (6)

Результати тестування правильних класів еквівалентності представлені в табл. 2.22.

Таблиця 2.22 – Тести правильних класів еквівалентності для тестування варіанту використання «Авторизація»

№ тесту	Покритий клас	Вхідні дані	Вихідні данні	Результат тестування
1	1, 4	Логін: Пароль:	Перехід на сторінку товарів	Тест пройдено

Результати тестування неправильних класів еквівалентності представлені в табл. 2.23.

Таблиця 2.23 – Тести неправильних класів еквівалентності для тестування варіанту використання «Авторизація»

№ тесту	Покритий клас	Вхідні дані	Вихідні данні	Результат тестування
1	2, 5	Логін: Пароль:	Повідомлення про помилку введення даних	Тест пройдено
2	3, 6	Логін: Пароль:	Повідомлення про помилку введення даних	Тест пройдено

Тестування неправильних класів еквівалентності для варіанту використання «Авторизація» було проведено успішно.

2.3.4 Випробування програмного забезпечення

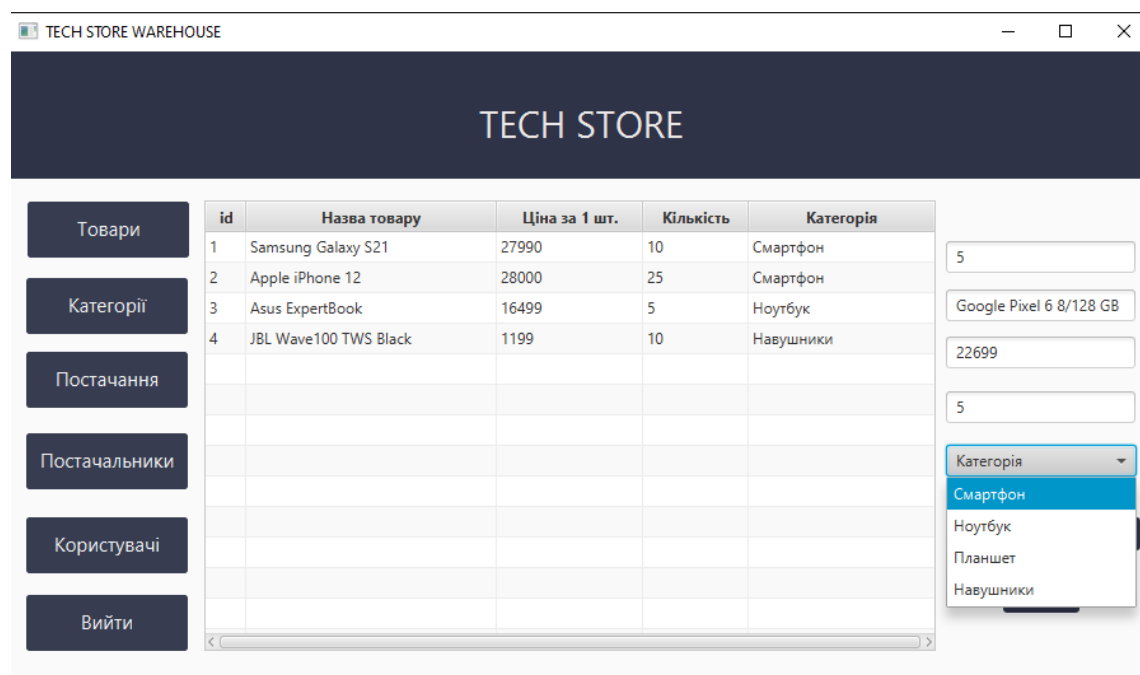
Для випробування був обраний варіант використання «Додавання товару до складу».

Для того, щоб додати товар до складу, потрібно відкрити вікно зі списком товарів та ввести у відповідні поля код товару, його назву, вартість за одиницю, кількість та категорію товару.

Вхідні дані:

- код товару;
- назва товару;
- вартість;
- кількість одиниць товару;
- категорія товару.

Очікуваним результатом буде додавання товару до списку. Додавання товару показане на рисунку 2.10.



TECH STORE WAREHOUSE

TECH STORE

Товари

Категорії

Постачання

Постачальники

Користувачі

Вийти

id	Назва товару	Ціна за 1 шт.	Кількість	Категорія
1	Samsung Galaxy S21	27990	10	Смартфон
2	Apple iPhone 12	28000	25	Смартфон
3	Asus ExpertBook	16499	5	Ноутбук
4	JBL Wave100 TWS Black	1199	10	Навушники

5

Google Pixel 6 8/128 GB

22699

5

Категорія

Смартфон

Ноутбук

Планшет

Навушники

Рисунок 2.10 – Додавання товару

Результат додавання показаний на рисунку 2.11.

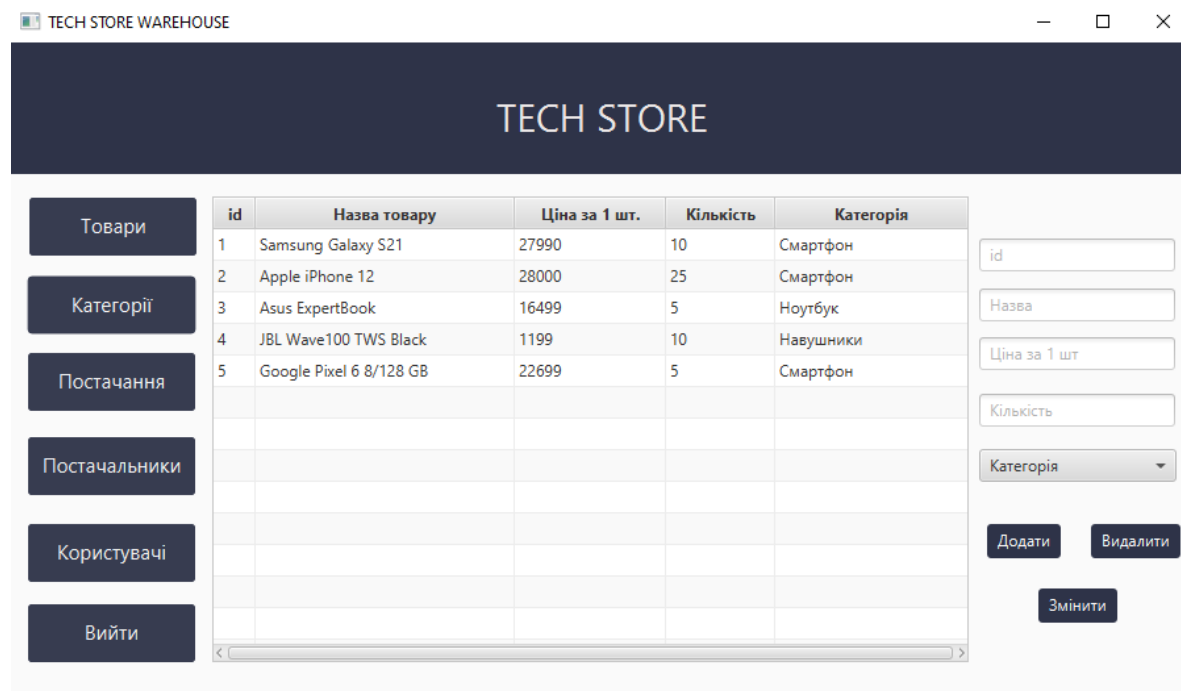


Рисунок 2.11 – Доданий товар на склад

Другим випробуванням буде введення неповних даних товару. Якщо не усі поля будуть заповнені, очікуваним результатом буде помилка і недодавання товару на склад. Результат показаний на рисунку 2.12.

TECH STORE WAREHOUSE

TECH STORE

Товари

Категорії

Постачання

Постачальники

Користувачі

Вийти

id	Назва товару	Ціна за 1 шт.	Кількість	Категорія
1	Samsung Galaxy S21	27990	10	Смартфон
2	Apple iPhone 12	28000	25	Смартфон
3	Asus ExpertBook	16499	5	Ноутбук
4	JBL Wave100 TWS Black	1199	10	Навушники

Додати
Видалити

Змінити

Рисунок 2.12 – Додавання товару з неповними даними

Після того, як ми ввели неповні дані, товар не був доданий на склад, тому що поля товару не повинні бути пустими. Результат на рисунку 2.13.

TECH STORE WAREHOUSE

TECH STORE

Товари

Категорії

Постачання

Постачальники

Користувачі

Вийти

id	Назва товару	Ціна за 1 шт.	Кількість	Категорія
1	Samsung Galaxy S21	27990	10	Смартфон
2	Apple iPhone 12	28000	25	Смартфон
3	Asus ExpertBook	16499	5	Ноутбук
4	JBL Wave100 TWS Black	1199	10	Навушники

Додати
Видалити

Змінити

Alert

Помилка

Дані введені некорректно!

OK

Рисунок 2.13 - Помилка при додаванні товару

Таким чином, було проведено випробування одного варіанту використання на коректні та некоректні дані.

3 РЕЗУЛЬТАТИ РОЗРОБКИ ПЗ ДЛЯ АВТОМАТИЗАЦІЇ УПРАВЛІННЯ СКЛАДОМ МАГАЗИНУ ЕЛЕКТРОНІКИ «TECH STORE»

В даній кваліфікаційній роботі було розроблене програмне забезпечення для автоматизації управління складом магазину електроніки «Tech Store». Програмне забезпечення у вигляді віконного додатку дозволяє користувачеві додавати, редагувати та видаляти дані про товари та постачальників. Адміністратор також може керувати користувачами, які користуються програмою.

В даній роботі був проведений аналіз роботи магазину, аналіз готових рішень, була здійснена постановка задачі та розроблено технічне завдання, що наведене в додатку А – «Технічне завдання».

Також, під час роботи над кваліфікаційною роботою, були виконані ескізний, технічний та робочий проекти.

В ескізному проекті була обрана мова моделювання, за допомогою якої були побудовані: контекстна діаграма, модель варіантів використання, концептуальна модель, ескіз інтерфейсу користувача.

У технічному проекті були розроблені: логічна модель БД, діаграма діяльності, послідовності, розгортання та діаграма класів.

У робочому проекті було обґрунтовано вибір інструментів розробки та СКБД, розроблена фізична модель БД, виконане тестування та випробування програмного забезпечення, котре показало працездатність та відповідність вимогам, які були представлені у технічному завданні.

Програмне забезпечення було написане на мові програмування Java у середовищі JetBrains IntelliJ IDEA. База даних була створена у СКБД MySQL Workbench. Приклад роботи програми, а саме вікно товарів, продемонстровано на рисунку 3.1.

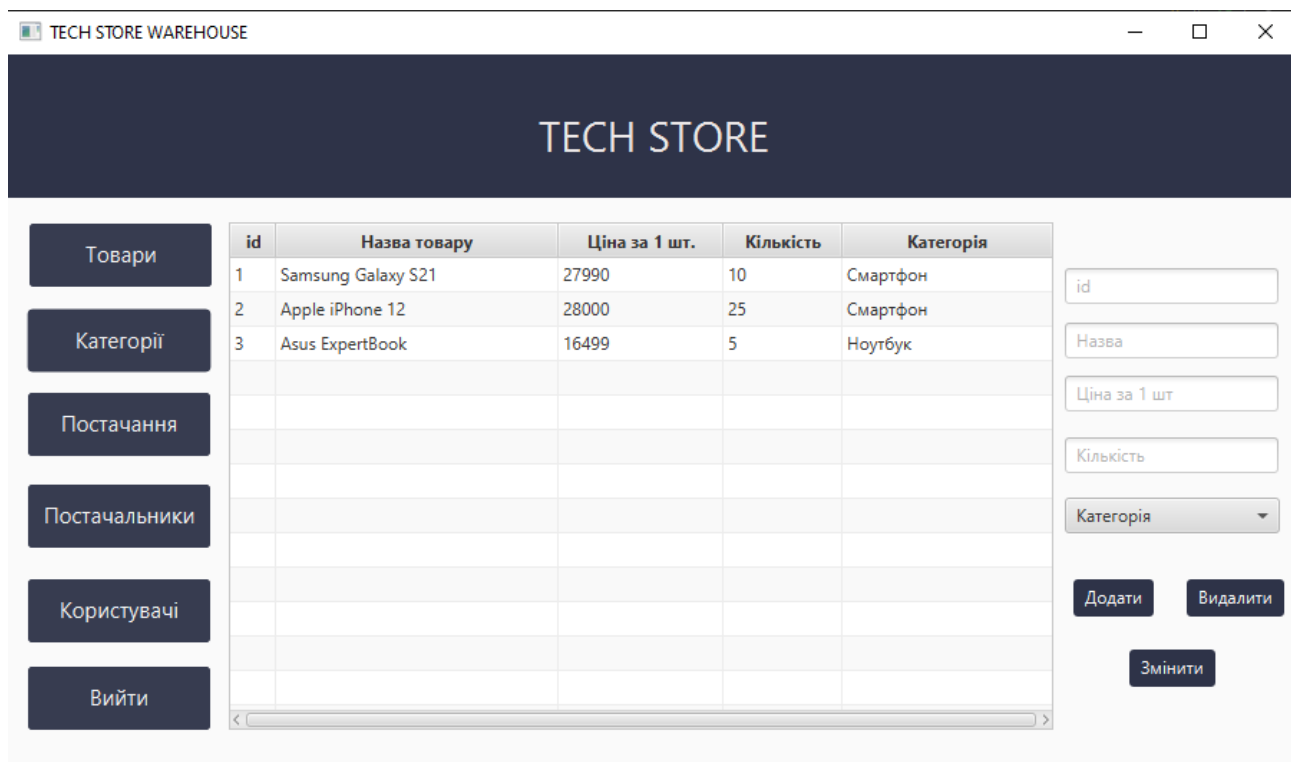


Рисунок 3.1 – Вікно товарів

Текст програми наведений у додатку Б – Текст програми.

Детальний опис програми наведений у додатку В – Опис програми.

Опис експлуатації даного програмного забезпечення був наведений у додатку Д – Інструкція користувача.

Програма та методика випробувань програмного забезпечення наведена у додатку Д – Програма та методика випробувань ПЗ.

4 ОХОРОНА ПРАЦІ

4.1 Вступ

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності [9].

Мета охорони праці – забезпечення безпечних, нешкідливих і сприятливих умов праці через вирішення багатьох складних завдань, основними з яких є:

- проектування підприємств, технологічних процесів і конструювання обладнання з обов'язковим виконанням вимог охорони праці;
- знаходження оптимальних співвідношень між різними факторами виробничого середовища, що дозволяє забезпечити мінімум несприятливого впливу їх на здоров'я працівників;
- встановлення, законодавче оформлення визначених норм кожного з несприятливих або небезпечних факторів, систематичний контроль за їх застосуванням;
- розробка конкретних заходів щодо покращення умов праці та забезпечення її безпеки на основі застосування у виробництві новітніх досягнень науки і техніки;
- застосування раціональних засобів захисту працівників від впливу несприятливих факторів виробничого середовища, а також втілення організаційних заходів, які нейтралізують або послаблюють ступінь їх впливу на організм людини;
- розробка та застосування методів і засобів оцінки ефективності заходів з охорони праці, що плануються і здійснюються.

Основним законом в галузі охорони праці є Закон України «Про охорону праці». Дія Закону України «Про охорону праці» поширюється на всіх юридичних та фізичних осіб, які відповідно до законодавства використовують найману працю, та на всіх працюючих [10].

4.2 Основні законодавчі і нормативні документи, які регламентують охорону праці в Україні

До основних документів, що регламентують охорону праці в Україні, відносяться:

- Конституція України;
- КЗпП;
- закони «Про охорону праці», «Про охорону здоров'я», «Про пожежну безпеку», «Про використання ядерної енергії та радіоактивний захист», «Про забезпечення санітарного та епідемічного благополуччя населення». Коло питань щодо охорони праці розглядається в підзаконних актах, указах Президента, постановах Верховної Ради та Кабміну, Цивільному, Кримінальному та Адміністративному кодексах України.

Кожній людині Конституція гарантує право на належні, безпечні і здорові умови праці. Конституцією гарантується захист від незаконного звільнення. Кожна людина має право на достатній життєвий рівень, на охорону здоров'я, медичну допомогу та медичне страхування.

В КЗпП охорона праці відображена в окремому розділі. В окремих статтях цього розділу розглянуто створення здорових і безпечних умов праці; додержання вимог охорони праці.

До найважливіших актів з охорони праці належать:

- Закон України «Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності» (23.09.1999 р. № 1105);
- Порядок розслідування та обліку нещасних випадків, професійних захворювань і аварій на виробництві (25.08.2004 р. №1112);
- Порядок видачі дозволів Державним комітетом з нагляду за охороною праці та його територіальними органами (15.10.2003 р. № 1631);
- Порядок проведення атестації робочих місць за умовами праці (01.08.1992 р. №442).

Спеціальними законодавчими актами є міжгалузеві та галузеві акти про охорону праці: Державні стандарти, Системи стандартів безпеки праці, Санітарні норми, норми радіаційної безпеки тощо.

4.3 Загальні вимоги до виробничих приміщень ЕОМ

Серед основних вимог виділяють [9]:

- розміщення робочих місць у підвальних і цокольних приміщеннях заборонено;
- площа на одне робоче місце ПК повинна бути не менше 6 м², а об'єм – не менше 20 м³;
- підлога всієї зони обслуговування ЕОМ має бути вкрита діелектричними килимами, з метою зниження електростатичного поля між оператором і ПК;
- усі металеві конструкції, що мають заземлення (батареї опалення, водопровідні труби, екрани електропроводки, тощо) повинні бути надійно захищені діелектричними щитками або сітками від випадкового дотику людини до них;

- в приміщеннях для роботи з ПК повинні бути кімнати для відпочинку і психологічного розвантаження працівників. У цих кімнатах повинно бути передбачена роздача тонізуючих напоїв, та місця для занять фізичною культурою.

Санітарно-гігієнічні вимоги до виробничого середовища ЕОМ визначають:

- мікроклімат;
- степінь освітленості;
- рівень шуму і вібрації;
- іонізуюче та неіонізуюче випромінювання.

Мікроклімат на робочих місцях з ПК нормується ДЕСТ 12.1.005-88 і повинен мати:

- температуру оточуючого повітря – 21–24^оС;
- відносну вологість – 40–60 %;
- швидкість руху повітря – 0,1–0,2 м/с.

Рівні шуму на робочих місцях ЕОМ визначені ДсанПіН 3.3.2-007-98 в залежності від його тональності та виду трудової діяльності користувачів ЕОМ (програмісти, оператори ЕОМ, чи оператори комп'ютерного набору). Рівні звукового тиску описані у таблиці 4.1.

Таблиця 4.1 – Рівні звукового тиску в дБ в октавних смугах частот, Гц.

Рівні звуку дБА/ дБАекв Вид трудової діяльності	31.5	63	125	250	500	1000	2000	4000	8000
Програмісти ЕОМ	71	61	54	49	45	42	40	38	50
Оператори ЕОМ та опер. комп'ю- терного набору	83	74	68	63	60	57	55	54	65
В приміщеннях, де розташову- ються шумні агрегати ЕОМ	91	83	77	73	70	68	66	64	75

Для нормування шуму застосовуються шумопоглинальні засоби, визначені спеціальними інженерно-акустичними розрахунками.

Освітлення на робочих місцях:

- Робочі місця з ПК повинні мати природне і штучне освітлення;
- Вікна приміщень з ПК повинні мати орієнтацію на північ або на північний схід, з КПО 1,5%. Дозволяється експлуатація ЕОМ у приміщеннях без природного освітлення за узгодженням з органами Держнаглядохоронпраці та санітарно-епідеміологічними установами;
- Рівень освітленості на робочому місці в зоні розміщення документів має бути 300–500 лк;
- Штучне освітлення виконується, як правило, люмінесцентними лампами. Допускається для місцевого освітлення використовувати лампи розжарювання;
- Місцеве освітлення не повинно створювати блисків на екрані ПК, а його освітленість повинна не перевищувати 300 лк.

Рівні вібрації не повинні перевищувати допустимих норм, визначених ДсанПіН 3.3.2-007-98.

Рівні іонізуючого випромінювання повинні відповідати НРБУ–97. Потужність експозиційної дози рентгенівського випромінювання на відстані 5 см від екрану ПК повинна бути не більше 0,1 мбер/год (100 мкР/год).

Рівні неіонізуючого випромінювання повинні відповідати вимогам ДЕСТ 12.1.006, СН №3206-85 «Гранично допустимі рівні магнітного поля частотою 50 Гц, СН №4557-88 «Санітарні норми ультрафіолетового випромінювання у виробничих приміщеннях».

4.4 Аналіз шкідливих та небезпечних факторів на робочому місці

Під час укладання трудових договорів роботодавець повинен поінформувати працівника під розписку про умови праці та про наявність на його робочому місці небезпечних і шкідливих виробничих факторів, які ще не усунуто, можливі наслідки їх впливу на здоров'я та про права працівника на пільги і компенсації за роботу в таких умовах відповідно до законодавства і колективного договору.

Небезпечний виробничий фактор – фактор середовища і трудового процесу, що може бути причиною гострого захворювання (отруєння), раптового різкого погіршення здоров'я або смерті.

Шкідливий виробничий фактор – фактор середовища або трудового процесу, вплив якого на працівника за певних умов (інтенсивність, тривалість дії тощо) може спричинити професійне або виробничо обумовлене захворювання, тимчасове або стійке зниження працездатності, підвищення частоти соматичних та інфекційних захворювань, призвести до порушення здоров'я як працівника, так і його нащадків.

Тривала і інтенсивна робота на комп'ютері може стати джерелом важких професійних та звичайних захворювань таких як [10]:

- неправильна постава – це призводить до подальшого розвитку викривлення хребта сколіозу, лордозу, кіфозу, а як результат головні болі, болі в області шиї і всього хребта, болі в області таза. Неправильно положення ніг може привести до артрити (запалення суглобів), артрозу (деформації);

- тунельний синдром – найвідоміше захворювання людей працюють за комп'ютером. Воно ж синдром зап'ястного каналу. Тунельний синдром проявляється після кількох годин напруженої роботи за комп'ютером;

- геморой – хвороба аналогічна варикозного розширення вен. Причиною хвороби може бути застій крові у венах;

- розвиток короткозорості – через те, що екран монітора за контрастом вище, ніж навколишні об'єкти розвивається короткозорість;
- порушення фокусування – наслідком напруженої роботи за монітором є порушення фокусування, яка може бути викликана перенапруженням очних м'язів;
- сухість очей – через рефлекс, заснованого на тому, що при погляді на джерело світла очей починає менше моргати виникає «обсушування» рогівки ока яке призводить до очних болів.

4.5 Заходи щодо зменшення впливу шкідливих факторів

Працюючи за комп'ютером, рекомендуємо дотримуватися правил тривалості роботи, правильної постави, розміру шрифтів та зображень, вимог до приміщення тощо [11].

Принципи правильної роботи за комп'ютером:

- у робочому приміщенні (кімнаті), де встановлені комп'ютери, щодня потрібно виконувати вологе прибирання;
- приміщення, у якому знаходяться комп'ютери, потрібно провітрювати щогодини;
- після кожного часу роботи рекомендується робити десяти хвилинну перерву, яку зручно суміщати з провітрюванням. За будь-яких умов безперервна робота за комп'ютером для дорослої людини не повинна перевищувати двох годин. Під час перерви не варто читати або дивитися телевізор;
- необхідно постійно слідкувати за станом екрану монітора: він має бути чистим, без плям та пилу;
- потрібно слідкувати за поставою: ноги повинні твердо стояти на підлозі чи на спеціальній підставці; стегна повинні бути розташовані під прямим кутом

до тулуба, а гомілки – під прямим кутом до стегон; сидіти потрібно прямо або злегка нахилившись вперед; відстань від очей до екрана монітора – не менше 55- 60 см; центр екрану має знаходитися на рівні очей чи трохи нижче; рекомендується хоча б раз на день виконувати гімнастику для очей;

- щоб попередити „синдром сухого ока”, потрібно моргати кожні 3-5 секунд;

- не використовувати звичайний телевізор замість монітора;

- у процесі роботи рекомендується періодично (приблизно раз на 20-30 хвилин) переводити погляд з екрана на найбільш віддалений предмет у кімнаті, а ще краще – на віддалений об’єкт за вікном;

- якщо з’явилося відчуття втоми, напруження, сонливості, важкості в очах, потрібно припинити роботу та хоча б трохи відпочити.

Рекомендовані умови для роботи за комп’ютером:

- твердий стілець. край стільця не повинен тиснути на судини під колінами;

- відстань до монітора повинна бути 50-70 см;

- перерву в сидячій роботі, 15-20 хвилин кожні 1-2 години;

- правильне освітлення робочого місця. при слабкому світлі очі напружуються і болять;

- потрібно закривати очі для відпочинку. час від часу відводите очі вбік, щоб дати відпочити своєму зору.

ВИСНОВКИ

У результаті кваліфікаційної роботи було проведено розв'язання практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме розробку програмного забезпечення для автоматизації управління складом магазину електроніки «Tech Store», котре дозволить працівникам магазину спростити та пришвидшити свою роботу в управлінні бізнес-процесами складської роботи підприємства.

У результаті аналізу діяльності магазину електроніки «Tech Store» та існуючих програмних продуктів, було зроблено висновок про доцільність розробки нового програмного продукту для існуючого магазину. Задача програмного забезпечення – автоматизувати складську роботу магазину електроніки «Tech Store». Були поставлені задачі та виділені основні функції програмного продукту.

У результаті кваліфікаційної роботи були розроблені ескізний, технічний та робочий проекти, на основі яких було розроблено програмне забезпечення для автоматизації управління складом магазину електроніки «Tech Store».

У розділі з охорони праці були розглянуті основні положення про охорону праці та проаналізовані небезпечні і шкідливі фактори при роботі з екранними пристроями та правила безпечної роботи за комп'ютером.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Warehouse Management System. URL: <https://www.it.ua/knowledge-base/technology-innovation/warehouse-management-system-wms> (дата звернення: 25.04.2022).
2. 1С: Підприємство: докладний огляд програмного забезпечення. URL: <https://itez.com.ua/what-is-1c.html> (дата звернення: 27.04.2022).
3. WMS UIS – Система управління складом. URL: <https://uislab.com/products/wms/> (дата звернення: 27.04.2022).
4. Леоненков А.В. Об'єктно-орієнтований аналіз та проектування з використанням UML та IBM RationalRoseM.: Інтернет-Університет Інформаційних Технологій; БІНОМ. Лабораторія знань, 2006, 319 с. (дата звернення 06.06.2022).
5. Що таке концептуальна модель даних? – визначення з техопедії – Розвиток – 2022. URL: <https://uk.theastrologypage.com/conceptual-data-model> (дата звернення 07.06.2022).
6. Логічна модель даних – визначення з wikiwand. URL: <https://www.wikiwand.com/uk> (дата звернення 07.06.2022).
7. PHP + MySQL – Запорізький національний університет. URL: PHP + MySQL (znu.edu.ua) (дата звернення 13.06.2022).
8. Що таке MySQL WorkBench. URL: MySQL Workbench - CoderLessons.com (дата звернення 13.06.2022).
9. М.П.Гандзюк. Основи охорони праці // Підручник – К.: Каравела; Львів: Новий Світ – 2000, 2003. – 408с (дата звернення 23.04.2022).
10. Закон України про охорону праці. URL: <https://zakon.rada.gov.ua/laws/main/2694-12#Text> (дата звернення 23.04.2022).

11. Правила безпечної роботи на комп'ютері. URL: <https://pedcollege.kiev.ua/index.php/77-robota-koledzhu/okhorona-pratsi/589-pravyly-bezpechnoi-roboty-na-kompiuteri> (дата звернення 23.04.2022).

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ

ВСТУП

Найменування програмного додатку, що розробляється – «Система для автоматизації управління складом». Область застосування – магазин електроніки «Tech Store».

1 ПІДСТАВА ДЛЯ РОЗРОБКИ

Розробка програмного забезпечення виконується на підставі завдання на кваліфікаційну роботу, виданого кафедрою ПЗАС НУК імені адмірала Макарова.

2 ПРИЗНАЧЕННЯ РОЗРОБКИ

2.1 Експлуатаційне призначення

Експлуатаційним призначенням програмного забезпечення системи управління складом магазину електроніки «Tech Store» є автоматизація управління складськими процесами, котре дозволить працівникам магазину спростити та пришвидшити свою роботу в управлінні бізнес-процесами складської роботи підприємства.

2.3 Функціональне призначення

Функціональним призначенням ПЗ для автоматизації управління складом магазину електроніки «Tech Store» є автоматизація процесів:

- реєстрації та авторизації користувачів;
- створювання нової поставки товарів;
- додавання нового товару на склад;
- редагування товару на складі;
- видалення товару зі складу;
- додавання нової категорії;
- видалення категорії;
- редагування категорії;
- створення нового користувача;
- редагування даних про користувача;
- видалення користувача;
- переглядання усіх товарів на складі;

3 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вимоги до функціональних характеристик

3.1.1 Вимоги до складу виконуваних функцій

Програмний додаток повинен виконувати всі нижчеописані функції.

Авторизація користувачів

Авторизація користувачів в ПЗ для надання їм функціональних можливостей. Обов'язковими полями для авторизації є логін та пароль. У випадку введення правильного логіну та паролю буде відкрите відповідне автоматизоване робоче місце. При введення некоректних чи незареєстрованих даних користувач отримуватиме відповідне системне повідомлення.

Представлення даних

Дані, що задіяні у роботі програмного додатку представляються у вигляді наступних таблиць у БД системи: товар, категорія, постачальники, користувачі, замовлення.

Перегляд товарів, категорій, користувачів, постачальників

Для перегляду товарів потрібно обрати потрібну для перегляду таблицю.

Додавання нового товару, категорії, користувача, постачальника

Для додавання потрібно обрати певне вікно та ввести всі необхідні атрибути даної таблиці.

Редагування товару, категорії, користувача, постачальника

Для редагування потрібно обрати певний запис та замінити інформацію потрібних атрибутів.

Видалення товару, категорії, користувача, постачальника.

Для видалення потрібно обрати певне вікно та відповідний запис з БД системи.

Формування поставок товарів

Для формування поставок товарів потрібно обрати вікно з поставками та ввести всі необхідні атрибути.

3.1.2 Вимоги до організаційних вхідних та вихідних даних

Вхідні дані: введення даних здійснюється за допомогою пристроїв введення даних (клавіатура та миша). Дані або вводяться вручну, або обираються із представлених списків.

Вихідні дані: виведення даних здійснюється за допомогою пристроїв виведення даних (монітор).

3.2 Вимоги до надійності

ПЗ повинен функціонувати без збоїв при умові безперебійної роботи комп'ютера. При виникненні збоїв в роботі комп'ютера, відновлення нормальної роботи програмного продукту повинне проводитися після перезавантаження ПЗ. Для забезпечення надійності збереження інформації в БД системи повинна використовуватися СУБД, що забезпечуватиме цілісність і надійність транзакцій і нести відповідальність за цілісність інформації, занесеної в БД.

У випадку втрачання частини інформації у БД системи, ПЗ повинне продовжувати коректне функціонування. У випадку неможливого продовження коректної роботи має бути виведене відповідне системне повідомлення користувача ПЗ.

У випадку введення користувачем ПЗ некоректної інформації – введення некоректних даних чи даних, незареєстрованих в системі, повинно бути виведене відповідне повідомлення про помилку та надати користувачу можливість виправити відповідну помилку.

3.3 Вимоги до умов експлуатації

Необхідний рівень підготовки користувачів: мінімальні навички в користуванні комп'ютером, призначеного для роботи в закритому приміщенні.

3.4 Вимоги до складу та параметрів технічних засобів

Система повинна задовольняти вказаним вимогам на комп'ютері наступної мінімальної комплекції:

- CPU: Intel(R) Celeron(R) – 2.16 МГц;
- RAM: 2 GB;
- HDD: 10 GB вільного місця.

3.5 Вимоги до інформаційної та програмної сумісності

Для повноцінного функціонування системи необхідна ОС Windows версії не нижче 7 та наявність встановленого Java Runtime Environment (JRE 1.8).

3.6 Вимоги до маркування та пакування

Продукт буде упакований в електронний архів і мати найменування Warehouse_managment_system.rar та містити:

- виконуваний файл Warehouse_managment_system.jar;
- файл, який містить структуру БД Warehouse_managment_system.sql.

4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

Програмна документація повинна містити:

- технічне завдання;
- текст програми;
- опис програми;
- інструкцію користувача;
- програму і методику випробувань.

5 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ

Стадії та етапи розробки представлені в таблиці А.1.

Таблиця А.1 – Стадії та етапи розробки програмного забезпечення

Стадії розробки	Етапи робіт	Вміст робіт	Контрольні точки
1	2	3	4
Технічне завдання	Обумовлення необхідності розробки	Постановка задачі. Збір початкових матеріалів. Вибір і обумовлення критеріїв ефективності та якості розроблюваного ПЗ	02.03.2022 -15.03.2022
	Розробка та затвердження технічного завдання	Визначення вимог до ПЗ. Узгодження і затвердження технічного завдання.	16.03.2022 -31.03.2022
Ескізний проект	Розробка ескізного проекту	Попередня розробка структури вхідних та вихідних даних. Уточнення методів рішення задачі. Розробка загального опису алгоритму розв'язання задачі	01.04.2022 -15.04.2022
	Затвердження ескізного проекту	Розробка пояснювальної записки. Узгодження і затвердження ескізного проекту	16.04.2022 -27.04.2022

Продовження табл. А.1

1	2	3	4
Технічний проект	Розробка технічного проекту	Уточнення структури вхідних та вихідних даних. Розробка алгоритму рішення задачі. Розробка структури ПЗ	28.04.2022 -03.05.2022
	Затвердження технічного проекту	Розробка пояснювальної записки. Узгодження і затвердження технічного проекту	04.05.2022 - 06.05.2022
Робочий проект	Розробка ПЗ	Програмування та налагодження програми	07.05.2022 -10.05.2022
	Розробка програмної документації	Розробка програмної документації відповідно до вимог ГОСТ 19.101-77	11.05.2022 -13.05.2022
Стадії розробки	Етапи робіт	Вміст робіт	Контрольні точки
	Випробування ПЗ	Розробка, узгодження та затвердження порядку і методики випробувань. Проведення попередніх випробувань. Коригування ПЗ і програмної документації за результатами випробувань.	14.05.2022 -15.05.2022

6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙОМУ

Контроль за аналізом та проектуванням кожної окремої частини програмного забезпечення відбувається на кожному етапі з урахуванням вимог, визначених у технічному завданні. Кожна стадія розробки повинна бути представлена в зазначені строки та узгоджена із замовником.

Прийом проводиться відповідно до програми і методики випробувань і документується за допомогою протоколу проведення випробувань. У разі знаходження помилок під час прийому програмного забезпечення, складається акт про знайдені помилки, який підписується представниками замовника і розробника і затверджуються керівником організації-замовника та організації-розробника. Розробник повинен, на протязі не більше ніж 2 тижні, виправити зазначені помилки і оповістити замовника про повторне проведення перевірки.

ДОДАТОК Б – ТЕКСТ ПРОГРАМИ

Лістинг 1 – Макет вікна авторизації

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.PasswordField?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="400.0" prefWidth="500.0"
style="-fx-background-color: #2E3348;"
xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="com.example.warehousemanagementsystem.LoginPage">
    <children>
        <AnchorPane layoutY="109.0" prefHeight="291.0"
prefWidth="500.0" style="-fx-background-color: fafafa;">
            <children>
                <TextField fx:id="loginField" layoutX="123.0"
layoutY="55.0" prefHeight="31.0" prefWidth="255.0" promptText="Логин" />
                <Button fx:id="signUpButton" layoutX="195.0"
layoutY="201.0" mnemonicParsing="false" prefHeight="25.0"
prefWidth="111.0" text="Авторизация">
                    <font>
                        <Font size="15.0" />
                    </font>
                </Button>
                <PasswordField fx:id="passwordField" layoutX="123.0"
layoutY="115.0" prefHeight="31.0" prefWidth="255.0" promptText="Пароль"
/>
            </children>
        </AnchorPane>
        <Label layoutX="170.0" layoutY="35.0" text="TECH STORE"
textFill="#ebe7e7">
            <font>
                <Font size="29.0" />
            </font>
        </Label>
    </children>
</AnchorPane>

```

Лістинг 2 – Макет вікна товарів

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.ComboBox?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.effect.Blend?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="500.0" prefWidth="900.0"
style="-fx-background-color: #2E3348;"
xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="com.example.warehousemanagementsystem.ProductPage">
    <children>
        <AnchorPane layoutY="100.0" prefHeight="400.0"
prefWidth="900.0" style="-fx-background-color: fafafa;">
            <children>
                <Button fx:id="categoriesPage" layoutX="14.0"
layoutY="78.0" mnemonicParsing="false" onAction="#openCat" opacity="0.96"
prefHeight="43.0" prefWidth="127.0" style="-fx-background-color:
#2E3348;" text="Категории" textFill="#f5f5f5">
                    <effect>
                        <Blend />
                    </effect>
                    <font>
                        <Font size="15.0" />
                    </font>
                </Button>
                <Button fx:id="orderPage" layoutX="14.0"
layoutY="136.0" mnemonicParsing="false" onAction="#openOrder"
opacity="0.96" prefHeight="43.0" prefWidth="127.0" style="-fx-background-
color: #2E3348;" text="Поставки" textFill="#f5f5f5">
                    <effect>

```

```

        <Blend />
    </effect>
    <font>
        <Font size="15.0" />
    </font>
</Button>
    <Button      fx:id="customerPage"      layoutX="14.0"
layoutY="200.0"      mnemonicParsing="false"      onAction="#openCustomer"
opacity="0.96" prefHeight="43.0" prefWidth="127.0" style="-fx-background-
color: #2E3348;" text="Постачальники" textFill="#f5f5f5">
    <effect>
        <Blend />
    </effect>
    <font>
        <Font size="15.0" />
    </font>
</Button>
    <Button      fx:id="usersPage"      layoutX="14.0"
layoutY="266.0"      mnemonicParsing="false"      onAction="#openUsers"
opacity="0.96" prefHeight="43.0" prefWidth="127.0" style="-fx-background-
color: #2E3348;" text="Користувачі" textFill="#f5f5f5">
    <effect>
        <Blend />
    </effect>
    <font>
        <Font size="15.0" />
    </font>
</Button>
    <Button      fx:id="logOut"      layoutX="14.0"
layoutY="327.0"      mnemonicParsing="false"      onAction="#openSignUp"
opacity="0.96" prefHeight="43.0" prefWidth="127.0" style="-fx-background-
color: #2E3348;" text="Вийти" textFill="#f5f5f5">
    <effect>
        <Blend />
    </effect>
    <font>
        <Font size="15.0" />

```

```

        </font>
    </Button>
    <Button      fx:id="productPage"      layoutX="15.0"
layoutY="18.0" mnemonicParsing="false" opacity="0.96" prefHeight="43.0"
prefWidth="127.0" style="-fx-background-color: #2E3348;" text="Товари"
textFill="#f5f5f5">

        <effect>
            <Blend />
        </effect>
        <font>
            <Font size="15.0" />
        </font>
    </Button>
    <TableView      fx:id="productList"      editable="true"
layoutX="154.0" layoutY="17.0" prefHeight="354.0" prefWidth="512.0"
AnchorPane.bottomAnchor="29.0" AnchorPane.leftAnchor="154.0"
AnchorPane.rightAnchor="234.0" AnchorPane.topAnchor="17.0">
        <columns>
            <TableColumn prefWidth="32.0" text="id" />
            <TableColumn prefWidth="180.0" text="Назва товару"
/>

            <TableColumn prefWidth="72.0" text="Ціна за 1
шт." />

            <TableColumn prefWidth="64.0" text="Кількість" />
            <TableColumn prefWidth="163.0" text="Категорія"
/>

        </columns>
    </TableView>
    <TextField      fx:id="idTextField"      layoutX="737.0"
layoutY="49.0" />
    <Label layoutX="675.0" layoutY="53.0" prefHeight="17.0"
prefWidth="71.0" text="Id" />
    <TextField      fx:id="nameTextField"      layoutX="737.0"
layoutY="87.0" />
    <TextField      fx:id="costField"      layoutX="737.0"
layoutY="124.0" />

```

```

        <TextField      fx:id="quantityField"      layoutX="737.0"
layoutY="167.0" />
        <Label layoutX="675.0" layoutY="91.0" prefHeight="17.0"
prefWidth="71.0" text="Назва" />
        <Label      layoutX="675.0"      layoutY="128.0"
prefHeight="17.0" prefWidth="71.0" text="Ціна" />
        <Label      layoutX="675.0"      layoutY="171.0"
prefHeight="17.0" prefWidth="71.0" text="Кількість" />
        <Label      layoutX="675.0"      layoutY="213.0"
prefHeight="17.0" prefWidth="71.0" text="Категорія" />
        <Button      fx:id="addButton"      layoutX="675.0"
layoutY="266.0" mnemonicParsing="false" onAction="#addProduct" style="-
fx-background-color: #2E3348;" text="Додати" textFill="WHITE" />
        <Button      fx:id="editButton"      layoutX="746.0"
layoutY="266.0" mnemonicParsing="false" style="-fx-background-color:
#2E3348;" text="Змінити" textFill="WHITE" />
        <Button      fx:id="deleteButton"      layoutX="822.0"
layoutY="266.0" mnemonicParsing="false" style="-fx-background-color:
#2E3348;" text="Видалити" textFill="WHITE" />
        <ComboBox fx:id="categoriesListField" layoutX="737.0"
layoutY="209.0" prefWidth="150.0" />
    </children>
</AnchorPane>
<Label layoutX="370.0" layoutY="36.0" text="TECH STORE"
textFill="#ebe7e7">
    <font>
        <Font size="29.0" />
    </font>
</Label>
</children>
</AnchorPane>

```

Лістинг 3 – Функціонал вікна товарів

```

package com.example.warehousemanagementsystem;

import java.io.IOException;

```

```

import java.net.URL;
import java.util.ResourceBundle;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.stage.Stage;

import javax.sql.StatementEvent;
import java.sql.*;

public class ProductPage {

    private Stage stage;
    private Scene scene;

    @FXML
    private ResourceBundle resources;

    @FXML
    private URL location;

    @FXML
    private Button addButton;

    private ObservableList<String> options =
FXCollections.observableArrayList();

    @FXML
    private ComboBox categoriesListField = new ComboBox(options);

```

```
@FXML
private Button categoriesPage;

@FXML
private TextField costField;

@FXML
private Button customerPage;

@FXML
private Button deleteButton;

@FXML
private Button editButton;

@FXML
private TextField idTextField;

@FXML
private Button logOut;

@FXML
private TextField nameTextField;

@FXML
private Button orderPage;

@FXML
private TableView<Product> productList;

@FXML
private TableColumn<Product, String> colCat;

@FXML
private TableColumn<Product, Integer> colPrice;
```

```

@FXML
private TableColumn<Product, Integer> colId;

@FXML
private TableColumn<Product, String> colName;

@FXML
private TableColumn<Product, Integer> colQuant;

@FXML
private Button productPage;

@FXML
private TextField quantityField;

@FXML
private Button usersPage;

@FXML
private void select(ActionEvent e) {

}

static Connection connection = null;

public static ObservableList<Product> getDataProduct() {
    ObservableList<Product> list = null;
    try {
        connection
DriverManager.getConnection("jdbc:mysql://localhost:3306/warehouse",
"root", "password");
        list = FXCollections.observableArrayList();
        PreparedStatement preparedStatement
connection.prepareStatement("select * from Product ");
        ResultSet rs = preparedStatement.executeQuery();

        while (rs.next()) {

```

```

        list.add(new
Product(Integer.parseInt(rs.getString("idproduct")),
rs.getString("name"),          Integer.parseInt(rs.getString("price")),
Integer.parseInt(rs.getString("quantity")),
rs.getString("category_fk")));
    }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return list;
}

@FXML
private void addProduct(ActionEvent e) {
    try {
        connection
=
DriverManager.getConnection("jdbc:mysql://localhost:3306/warehouse",
"root", "password");
        PreparedStatement          add
=
connection.prepareStatement("insert into PRODUCT values(?,?,?,?,?)");

        add.setInt(1, Integer.parseInt(idTextField.getText()));
        add.setString(2, nameTextField.getText());
        add.setInt(3, Integer.parseInt(costField.getText()));
        add.setInt(4,
Integer.parseInt(quantityField.getText()));
        add.setString(5,
categoriesListField.getValue().toString());

        int row = add.executeUpdate();
        connection.close();
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

```

```

@FXML
private void deleteProduct(ActionEvent e) {
    try {
        connection
        DriverManager.getConnection("jdbc:mysql://localhost:3306/warehouse",
        "root", "password");

        String id = idTextField.getText();
        String query = "delete from Product where
idproduct="+id;
        Statement add = connection.createStatement();
        add.executeUpdate(query);

        connection.close();
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

@FXML
void Select(ActionEvent event) {

    String s
categoriesListField.getSelectionModel().getSelectedItem().toString();
}

public void fillComboBox() {
    try {
        String query = "select name from Category";
        ps = connection.prepareStatement(query);
        rs = ps.executeQuery();

        while (rs.next()) {
            options.add(rs.getString("name"));
        }

        ps.close();
    }
}

```

```

        ps.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

ObservableList<Product> listProd;

int index = -1;

ResultSet rs = null;
PreparedStatement ps = null;

@FXML
void initialize() {
    fillComboBox();
    colId.setCellValueFactory(new PropertyValueFactory<Product,
Integer>("id"));
    colName.setCellValueFactory(new
PropertyValueFactory<Product, String>("name"));
    colPrice.setCellValueFactory(new
PropertyValueFactory<Product, Integer>("price"));
    colQuant.setCellValueFactory(new
PropertyValueFactory<Product, Integer>("quantity"));
    colCat.setCellValueFactory(new
PropertyValueFactory<Product, String>("category"));

    listProd = getDataProduct();
    productList.setItems(listProd);
}

@FXML
private void openCat(ActionEvent e) throws IOException {
    Parent root =
FXMLLoader.load((getClass().getResource("CategoriesPage.fxml")));
    stage =
(Stage)((Node)e.getSource()).getScene().getWindow();

```

```

        scene = new Scene(root, 900, 500);
        stage.setScene(scene);
        stage.show();
    }

    @FXML
    private void openOrder(ActionEvent e) throws IOException {
        Parent root =
FXMLLoader.load((getClass().getResource("OrderPage.fxml")));
        stage =
(Stage)((Node)e.getSource()).getScene().getWindow();
        scene = new Scene(root, 900, 500);
        stage.setScene(scene);
        stage.show();
    }

    @FXML
    private void openCustomer(ActionEvent e) throws IOException {
        Parent root =
FXMLLoader.load((getClass().getResource("CustomersPage.fxml")));
        stage =
(Stage)((Node)e.getSource()).getScene().getWindow();
        scene = new Scene(root, 900, 500);
        stage.setScene(scene);
        stage.show();
    }

    @FXML
    private void openUsers(ActionEvent e) throws IOException {
        Parent root =
FXMLLoader.load((getClass().getResource("UsersPage.fxml")));
        stage =
(Stage)((Node)e.getSource()).getScene().getWindow();
        scene = new Scene(root, 900, 500);
        stage.setScene(scene);
        stage.show();
    }

```

```

@FXML
private void openSignUp(ActionEvent e) throws IOException {
    Parent root =
FXMLLoader.load((getClass().getResource("LoginPage.fxml")));
    stage =
(Stage)((Node)e.getSource()).getScene().getWindow();
    scene = new Scene(root, 900, 500);
    stage.setScene(scene);
    stage.show();
}
}

```

Лістинг 4 – Макет вікна категорій

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.effect.Blend?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="500.0" prefWidth="900.0"
style="-fx-background-color: #2E3348;"
xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="com.example.warehousemanagementsystem.CategoriesPage">
    <children>
        <AnchorPane layoutY="100.0" prefHeight="400.0"
prefWidth="900.0" style="-fx-background-color: fafafa;">
            <children>
                <Button fx:id="categoriesPage" layoutX="14.0"
layoutY="78.0" mnemonicParsing="false" opacity="0.96" prefHeight="43.0"
prefWidth="127.0" style="-fx-background-color: #2E3348;" text="Категорії"
textFill="#f5f5f5">
                    <effect>
                        <Blend />
                    </effect>
                    <font>
                        <Font size="15.0" />
                    </font>
                </Button>
            </children>
        </AnchorPane>
    </children>
</AnchorPane>

```

```

        <Button          fx:id="orderPage"          layoutX="14.0"
layoutY="136.0" mnemonicParsing="false" opacity="0.96" prefHeight="43.0"
prefWidth="127.0"          style="-fx-background-color: #2E3348;"
text="Постачання" textFill="#f5f5f5">
        <effect>
            <Blend />
        </effect>
        <font>
            <Font size="15.0" />
        </font>
    </Button>
    <Button          fx:id="customerPage"          layoutX="14.0"
layoutY="200.0" mnemonicParsing="false" opacity="0.96" prefHeight="43.0"
prefWidth="127.0"          style="-fx-background-color: #2E3348;"
text="Постачальники" textFill="#f5f5f5">
        <effect>
            <Blend />
        </effect>
        <font>
            <Font size="15.0" />
        </font>
    </Button>
    <Button          fx:id="usersPage"          layoutX="14.0"
layoutY="266.0" mnemonicParsing="false" opacity="0.96" prefHeight="43.0"
prefWidth="127.0"          style="-fx-background-color: #2E3348;"
text="Користувачі" textFill="#f5f5f5">
        <effect>
            <Blend />
        </effect>
        <font>
            <Font size="15.0" />
        </font>
    </Button>
    <Button          fx:id="logOut"          layoutX="14.0"
layoutY="327.0" mnemonicParsing="false" opacity="0.96" prefHeight="43.0"
prefWidth="127.0" style="-fx-background-color: #2E3348;" text="Вийти"
textFill="#f5f5f5">
        <effect>
            <Blend />
        </effect>
        <font>
            <Font size="15.0" />
        </font>
    </Button>
    <Button          fx:id="productPage"          layoutX="15.0"
layoutY="18.0" mnemonicParsing="false" opacity="0.96" prefHeight="43.0"
prefWidth="127.0" style="-fx-background-color: #2E3348;" text="Товари"
textFill="#f5f5f5">
        <effect>
            <Blend />
        </effect>
        <font>
            <Font size="15.0" />
        </font>
    </Button>

```

```

        <TableView fx:id="categorytList" editable="true"
layoutX="154.0" layoutY="17.0" prefHeight="354.0" prefWidth="512.0"
AnchorPane.bottomAnchor="29.0" AnchorPane.leftAnchor="154.0"
AnchorPane.rightAnchor="234.0" AnchorPane.topAnchor="17.0">
        <columns>
        <TableColumn prefWidth="55.0" text="id" />
        <TableColumn
prefWidth="456.0"
text="Категорія товару" />
        </columns>
        </TableView>
        <TextField fx:id="idTextField" layoutX="737.0"
layoutY="49.0" />
        <Label layoutX="675.0" layoutY="53.0"
prefHeight="17.0" prefWidth="71.0" text="Id" />
        <TextField fx:id="nameTextField" layoutX="737.0"
layoutY="87.0" />
        <Label layoutX="675.0" layoutY="91.0"
prefHeight="17.0" prefWidth="71.0" text="Назва" />
        <Button fx:id="addButton" layoutX="675.0"
layoutY="145.0" mnemonicParsing="false" style="-fx-background-color:
#2E3348;" text="Додати" textFill="WHITE" />
        <Button fx:id="editButton" layoutX="746.0"
layoutY="145.0" mnemonicParsing="false" style="-fx-background-color:
#2E3348;" text="Змінити" textFill="WHITE" />
        <Button fx:id="deleteButton" layoutX="818.0"
layoutY="145.0" mnemonicParsing="false" style="-fx-background-color:
#2E3348;" text="Видалити" textFill="WHITE" />
        </children>
    </AnchorPane>
    <Label layoutX="370.0" layoutY="36.0" text="TECH STORE"
textFill="#ebe7e7">
        <font>
        <Font size="29.0" />
        </font>
    </Label>
</children>
</AnchorPane>

```

Лістинг 5 – Функціонал вікна категорій

```

package com.example.warehousemanagementsystem;

import java.io.IOException;
import java.net.URL;
import java.sql.*;
import java.util.ResourceBundle;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;

```

```

import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.stage.Stage;

public class CategoriesPage {

    private Stage stage;
    private Scene scene;
    @FXML
    private ResourceBundle resources;
    @FXML
    private URL location;

    @FXML
    private Button addButton;
    @FXML
    private Button categoriesPage;
    @FXML
    private TableView<Category> categoryList;
    @FXML
    private TableColumn<Category, String> colCat;
    @FXML
    private TableColumn<Category, Integer> colId;
    @FXML
    private Button customerPage;
    @FXML
    private Button deleteButton;
    @FXML
    private Button editButton;
    @FXML
    private TextField idTextField;
    @FXML
    private Button logOut;
    @FXML
    private TextField nameTextField;
    @FXML
    private Button orderPage;
    @FXML
    private Button productPage;
    @FXML
    private Button usersPage;

    static Connection connection = null;

    public static ObservableList<Category> getDataProduct() {
        ObservableList<Category> list = null;
        try {
            connection
DriverManager.getConnection("jdbc:mysql://localhost:3306/warehouse",
"root", "password");
            list = FXCollections.observableArrayList();

```

```

        PreparedStatement preparedStatement =
connection.prepareStatement("select * from Category ");
        ResultSet rs = preparedStatement.executeQuery();

        while (rs.next()) {
            list.add(new
Category(Integer.parseInt(rs.getString("idcategory")),
rs.getString("name")));
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return list;
}

@FXML
private void addCategory(ActionEvent e) {
    try {
        connection
DriverManager.getConnection("jdbc:mysql://localhost:3306/warehouse",
"root", "password");
        PreparedStatement add
connection.prepareStatement("insert into Category values(?,?)");

        add.setInt(1, Integer.parseInt(idTextField.getText()));
        add.setString(2, nameTextField.getText());

        int row = add.executeUpdate();
        connection.close();
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

@FXML
private void deleteCategory(ActionEvent e) {
    try {
        connection
DriverManager.getConnection("jdbc:mysql://localhost:3306/warehouse",
"root", "password");

        String id = idTextField.getText();
        String query = "delete from Category where
idproduct="+id;
        Statement add = connection.createStatement();
        add.executeUpdate(query);

        connection.close();
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

ObservableList<Category> listCat;

int index = -1;

```

```

        ResultSet rs = null;
        PreparedStatement ps = null;

        @FXML
        void initialize() {
            colId.setCellValueFactory(new
PropertyValuFactory<Category, Integer>("idCat"));
            colCat.setCellValueFactory(new
PropertyValuFactory<Category, String>("name"));

            listCat = getDataProduct();
            categoryList.setItems(listCat);
        }

        @FXML
        private void openProd(ActionEvent e) throws IOException {
            Parent root
FXMLLoader.load((getClass().getResource("ProductPage.fxml")));
            stage
            (Stage)((Node)e.getSource()).getScene().getWindow();
            scene = new Scene(root, 900, 500);
            stage.setScene(scene);
            stage.show();
        }

        @FXML
        private void openOrder(ActionEvent e) throws IOException {
            Parent root
FXMLLoader.load((getClass().getResource("OrderPage.fxml")));
            stage
            (Stage)((Node)e.getSource()).getScene().getWindow();
            scene = new Scene(root, 900, 500);
            stage.setScene(scene);
            stage.show();
        }

        @FXML
        private void openCustomer(ActionEvent e) throws IOException {
            Parent root
FXMLLoader.load((getClass().getResource("CustomerPage.fxml")));
            stage
            (Stage)((Node)e.getSource()).getScene().getWindow();
            scene = new Scene(root, 900, 500);
            stage.setScene(scene);
            stage.show();
        }

        @FXML
        private void openUsers(ActionEvent e) throws IOException {
            Parent root
FXMLLoader.load((getClass().getResource("UsersPage.fxml")));
            stage
            (Stage)((Node)e.getSource()).getScene().getWindow();
            scene = new Scene(root, 900, 500);
            stage.setScene(scene);

```

```

        stage.show();
    }

    @FXML
    private void openSignUp(ActionEvent e) throws IOException {
        Parent root
FXMLLoader.load((getClass().getResource("LoginPage.fxml")));
        stage
        (Stage)((Node)e.getSource()).getScene().getWindow();
        scene = new Scene(root, 900, 500);
        stage.setScene(scene);
        stage.show();
    }
}

```

Лістинг 6 – Макет вікна постачальників

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.effect.Blend?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="500.0" prefWidth="900.0"
style="-fx-background-color: #2E3348;"
xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="com.example.warehousemanagementsystem.CustomersPage">
    <children>
        <AnchorPane layoutY="100.0" prefHeight="400.0"
prefWidth="900.0" style="-fx-background-color: fafafa;">
            <children>
                <Button fx:id="categoriesPage" layoutX="14.0"
layoutY="78.0" mnemonicParsing="false" opacity="0.96" prefHeight="43.0"
prefWidth="127.0" style="-fx-background-color: #2E3348;" text="Категорії"
textFill="#f5f5f5">
                    <effect>
                        <Blend />
                    </effect>
                    <font>
                        <Font size="15.0" />
                    </font>
                </Button>
                <Button fx:id="orderPage" layoutX="14.0"
layoutY="136.0" mnemonicParsing="false" opacity="0.96" prefHeight="43.0"

```

```

prefWidth="127.0" style="-fx-background-color: #2E3348;"
text="Постачання" textFill="#f5f5f5">
    <effect>
        <Blend />
    </effect>
    <font>
        <Font size="15.0" />
    </font>
</Button>
<Button fx:id="customerPage" layoutX="14.0"
layoutY="200.0" mnemonicParsing="false" opacity="0.96" prefHeight="43.0"
prefWidth="127.0" style="-fx-background-color: #2E3348;"
text="Постачальники" textFill="#f5f5f5">
    <effect>
        <Blend />
    </effect>
    <font>
        <Font size="15.0" />
    </font>
</Button>
<Button fx:id="usersPage" layoutX="14.0"
layoutY="266.0" mnemonicParsing="false" opacity="0.96" prefHeight="43.0"
prefWidth="127.0" style="-fx-background-color: #2E3348;"
text="Користувачі" textFill="#f5f5f5">
    <effect>
        <Blend />
    </effect>
    <font>
        <Font size="15.0" />
    </font>
</Button>
<Button fx:id="logOut" layoutX="14.0"
layoutY="327.0" mnemonicParsing="false" opacity="0.96" prefHeight="43.0"
prefWidth="127.0" style="-fx-background-color: #2E3348;" text="Вийти"
textFill="#f5f5f5">
    <effect>
        <Blend />
    </effect>
    <font>
        <Font size="15.0" />
    </font>
</Button>
<Button fx:id="productPage" layoutX="15.0"
layoutY="18.0" mnemonicParsing="false" opacity="0.96" prefHeight="43.0"
prefWidth="127.0" style="-fx-background-color: #2E3348;" text="Товари"
textFill="#f5f5f5">
    <effect>
        <Blend />
    </effect>
    <font>
        <Font size="15.0" />
    </font>
</Button>
<TableView fx:id="customersList" editable="true"
layoutX="154.0" layoutY="17.0" prefHeight="354.0" prefWidth="573.0"

```

```

AnchorPane.bottomAnchor="29.0"                AnchorPane.leftAnchor="154.0"
AnchorPane.rightAnchor="173.0" AnchorPane.topAnchor="17.0">
    <columns>
        <TableColumn prefWidth="45.0" text="id" />
        <TableColumn prefWidth="176.0" text="Назва
постачальника" />
        <TableColumn                                prefWidth="210.0"
text="Адреса" />
        <TableColumn                                prefWidth="141.0"
text="Телефон" />
    </columns>
    </TableView>
    <TextField fx:id="idTextField" layoutX="737.0"
layoutY="49.0" promptText="id" />
    <TextField fx:id="nameField" layoutX="737.0"
layoutY="87.0" promptText="Назва" />
    <TextField fx:id="adressField" layoutX="737.0"
layoutY="124.0" promptText="Адреса" />
    <TextField fx:id="phoneField" layoutX="737.0"
layoutY="167.0" promptText="Телефон" />
    <Button fx:id="addButton" layoutX="737.0"
layoutY="231.0" mnemonicParsing="false" style="-fx-background-color:
#2E3348;" text="Додати" textFill="WHITE" />
    <Button fx:id="editButton" layoutX="782.0"
layoutY="275.0" mnemonicParsing="false" style="-fx-background-color:
#2E3348;" text="Змінити" textFill="WHITE" />
    <Button fx:id="deleteButton" layoutX="818.0"
layoutY="231.0" mnemonicParsing="false" style="-fx-background-color:
#2E3348;" text="Видалити" textFill="WHITE" />
    </children>
</AnchorPane>
<Label layoutX="370.0" layoutY="36.0" text="TECH STORE"
textFill="#ebe7e7">
    <font>
        <Font size="29.0" />
    </font>
</Label>
</children>
</AnchorPane>

```

ДОДАТОК В – ОПИС ПРОГРАМИ

1 ЗАГАЛЬНІ ВІДОМОСТІ

1.1 Позначення і найменування програми

Найменування – «Розробка програмного забезпечення для автоматизації управління складом магазину електроніки «Tech Store»», що розробляється на підставі завдання на кваліфікаційну роботу, виданого кафедрою ПЗАС НУК імені адмірала Макарова. Позначення програми - «Система управління складом».

1.2 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення потребує від пристрою наступних характеристик:

- мінімальна роздільна здатність екрана користувача: 320x568;
- оперативна пам'ять: 1гб;
- клавіатура, миша або сумісний вказівний пристрій;
- процесор: 300 МГц.

1.3 Мова програмування

Для реалізації проекту була обрана мова програмування Java. Середовищем розробки була обрана JetBrains IntelliJ IDEA.

2 ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

2.1 Функціональне призначення

Функціональним призначенням програмного забезпечення інтернет-аукціону є автоматизація наступних процесів для адміністратора:

- додавати нового користувача в систему;
- редагувати користувача;
- видаляти користувача;
- створювати нове замовлення товарів;
- додавати новий товар на склад;
- редагувати дані про товар;
- видаляти товар;
- переглядати усі товари на складі;
- додавати нового постачальника;
- редагувати постачальника;
- видаляти постачальника;
- додавати нову категорію товарів;
- редагувати категорію;
- видаляти категорію товарів.

Для працівника складу мають бути реалізовані наступні функції:

- створювати нове замовлення товарів;
- додавати новий товар на склад;
- редагувати дані про товар;
- видаляти товар;
- переглядати усі товари на складі;
- додавати нового постачальника;

- редагувати постачальника;
- видаляти постачальника;
- додавати нову категорію товарів;
- редагувати категорію;
- видаляти категорію товарів.

3 ОПИС ЛОГІЧНОЇ СТРУКТУРИ

3.1 Структура програми

Програмний забезпечення «Система для автоматизації управління складом магазину електроніки «Tech Store»» має дві складові частини: клієнтська частина – це графічний інтерфейс та база даних – програмне забезпечення, що забезпечує зберігання даних та їх видачу в момент запиту.

Логічна структура програми представлена взаємодією основних модулів програми у вигляді діаграми варіантів використання.

3.4 Алгоритм програми

Робота програмного забезпечення системи управління складом представляє взаємодією між клієнтською частиною та базою даних. Алгоритм роботи програми є наступним:

- Користувач взаємодіє з програмним забезпеченням системи управління складом;
- Програмне забезпечення звертається до бази даних (якщо це потрібно) та відсилає результат;
- Програмне забезпечення відображає отриманий результат.

3.3 Використовувані методи

Додаток був розроблений за допомогою об'єктно-орієнтованої мови програмування Java.

4 ТЕХНІЧНІ ЗАСОБИ, ЩО ВИКОРИСТОВУЮТЬСЯ

Програмне забезпечення для автоматизації управління складом потребує від пристрою, з якого буде переглядатися, наступних мінімальних характеристик:

- мінімальна роздільна здатність екрана користувача: 320x568;
- оперативна пам'ять: 1Гб;
- клавіатура, миша або сумісний вказівний пристрій;
- процесор: 300 МГц.

5 ВИКЛИК ТА ЗАВАНТАЖЕННЯ

Для завантаження додатку необхідно встановити на комп'ютері Java Runtime Environment. Після відкриття вікна авторизації, користувач повинен ввести дані для входу у додаток.

6 ВХІДНІ ДАНІ

Вхідні дані: введення даних здійснюється за допомогою пристроїв введення даних (клавіатура комп'ютера або іншого пристрою, миша). Дані програмного забезпечення вводяться вручну.

7 ВИХІДНІ ДАНІ

Вихідні дані: виведення даних здійснюється на електронний пристрій для відображення інформації, наприклад, монітор комп'ютера.

ДОДАТОК Г – ІНСТРУКЦІЯ КОРИСТУВАЧА

1 МОДУЛЬ КОРИСТУВАЧА

Для того, щоб запустити додаток, на комп'ютері повинно бути встановлено Java Runtime Environment, яке складається з Java Virtual Machine (JVM) базових класів платформи Java та допоміжних бібліотек платформи Java. Приклад нового встановлення Java, показаний на рисунку Г.1.



Рисунок Г.1 – Вікно встановлення Java

Після встановлення, можна працювати з Java-додатками.

2 ВХІД У ДОДАТОК

Щоб запустити додаток, користувачу потрібно клацнути 2 рази на значок програми Warehouse_managment_system.jar. Після цього з'явиться сторінка з авторизацією. Вікно авторизації показано на рисунку Г.2.

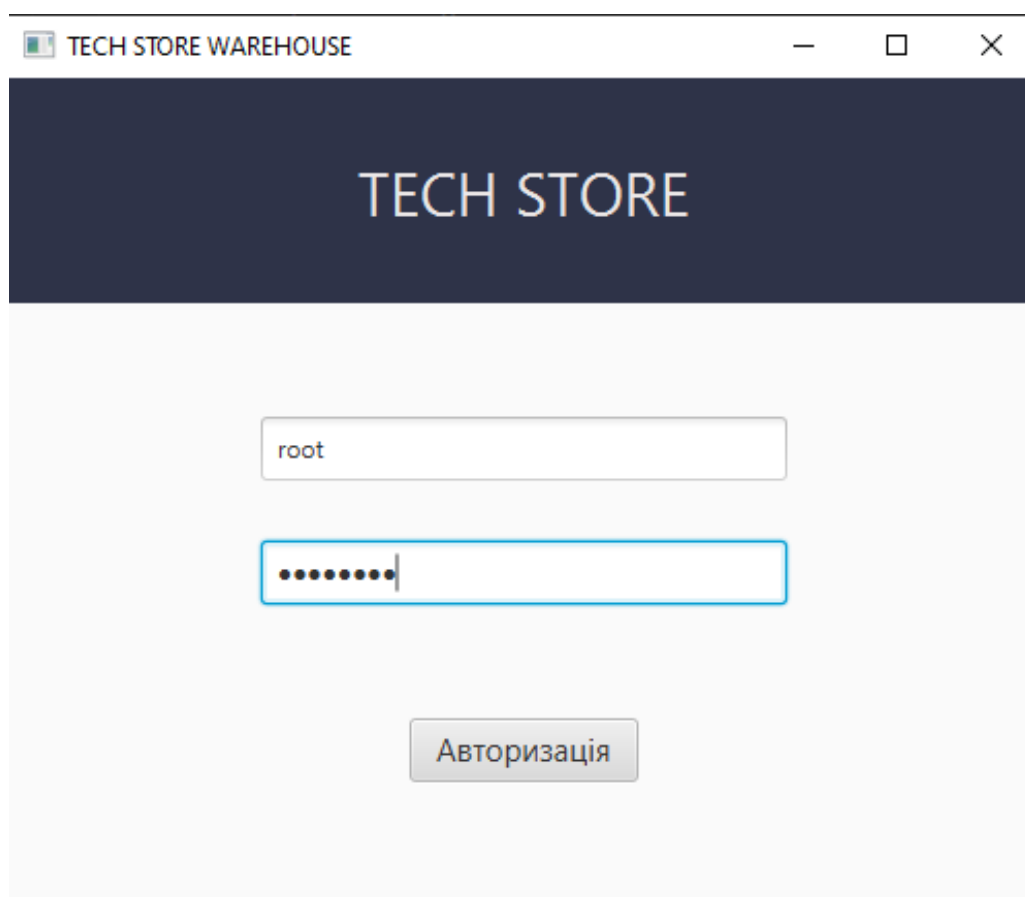
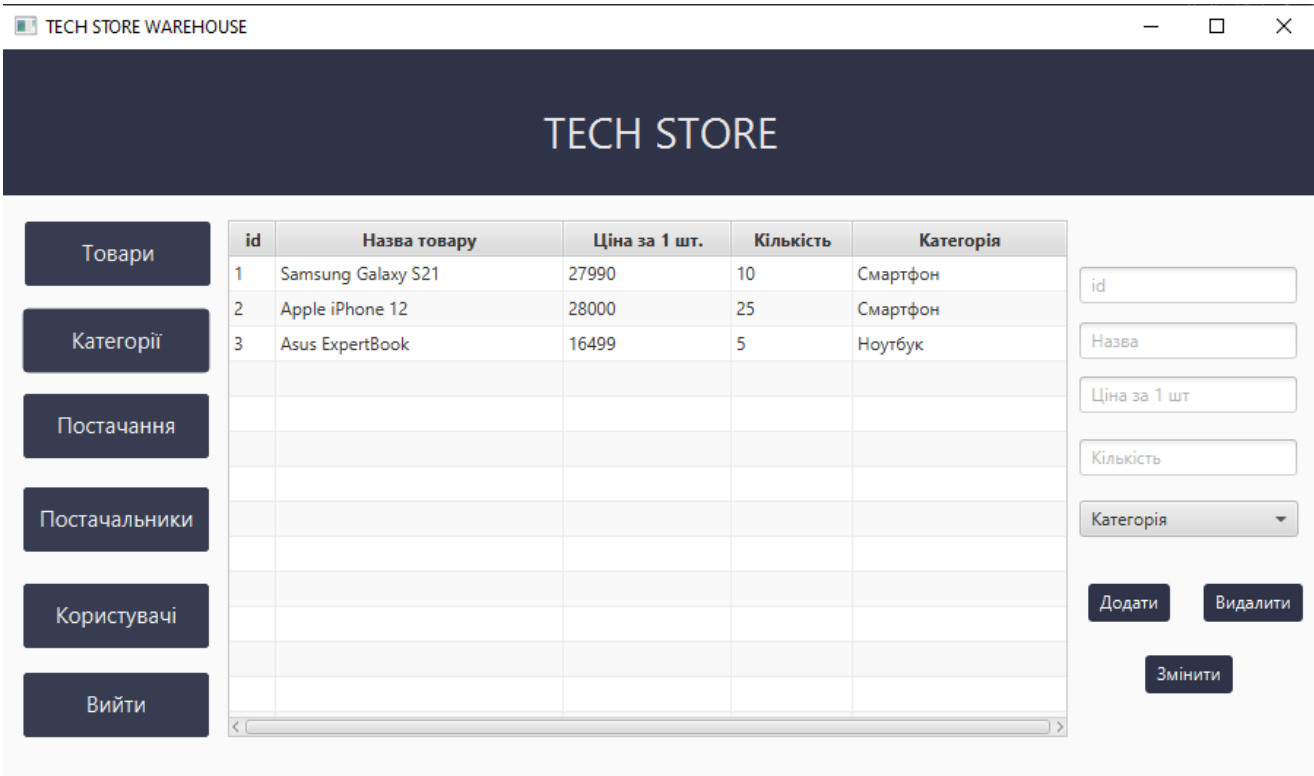


Рисунок Г.2 – Вікно авторизації

Якщо введені дані будуть некоректними, буде помилка, і програма попросить знову ввести дані. Після того, як дані були введені правильні, користувач потрапить у вікно з товарами.

3 СТОРІНКА ТОВАРІВ

Сторінка с товарами представляє собою вікно, де зліва є кнопки, натиснувши на які, можна перейти на інші вікна програми. По центру вікна є велика таблиця, де є повний список товарів, код товару, назва, ціна за одиницю товару, кількість товарів, які є на складі, а також категорія цього товару. Праворуч від таблиці є поля, які відповідають полям товару у таблиці, а також кнопки з додаванням товару, видаленням та редагуванням. Для того, щоб ввести новий товар до складу, потрібно заповнити усі відповідні поля та натиснути на кнопку «Додати». Для видалення товару потрібно ввести код товару, який існує на складі, та натиснути кнопку «Видалити». Для редагування, потрібно ввести код існуючого на складі товару, вписати у інші поля ті дані, які потрібні, та натиснути на кнопку «Змінити». Вікно товарів показано на рисунку Г.3.



The screenshot shows a web application window titled "TECH STORE WAREHOUSE". The main header is "TECH STORE". On the left, there is a sidebar with buttons: "Товари", "Категорії", "Постачання", "Постачальники", "Користувачі", and "Вийти". The main area displays a table with the following data:

id	Назва товару	Ціна за 1 шт.	Кількість	Категорія
1	Samsung Galaxy S21	27990	10	Смартфон
2	Apple iPhone 12	28000	25	Смартфон
3	Asus ExpertBook	16499	5	Ноутбук

On the right side of the table, there are input fields for "id", "Назва", "Ціна за 1 шт", "Кількість", and a dropdown for "Категорія". Below these fields are three buttons: "Додати", "Видалити", and "Змінити".

Рисунок Г.3 – Вікно з товарами

З цього вікна, користувач може перейти на інше вікно за вибором. Одним з інших вікон є вікно категорій.

4 ВІКНО КАТЕГОРІЙ

Вікно категорій схоже на попереднє, з однією відмінністю – тут по центру вікна розташована таблиця зі списком категорій товару. Спосіб додавання, видалення та редагування категорії ідентичний до того, як це робиться у вікні з товарами. Тут менше полів для вводу даних, так як в таблиці категорій є лише 2 поля – код та назва категорії. Вікно зі списком категорій показано на рисунку Г.4

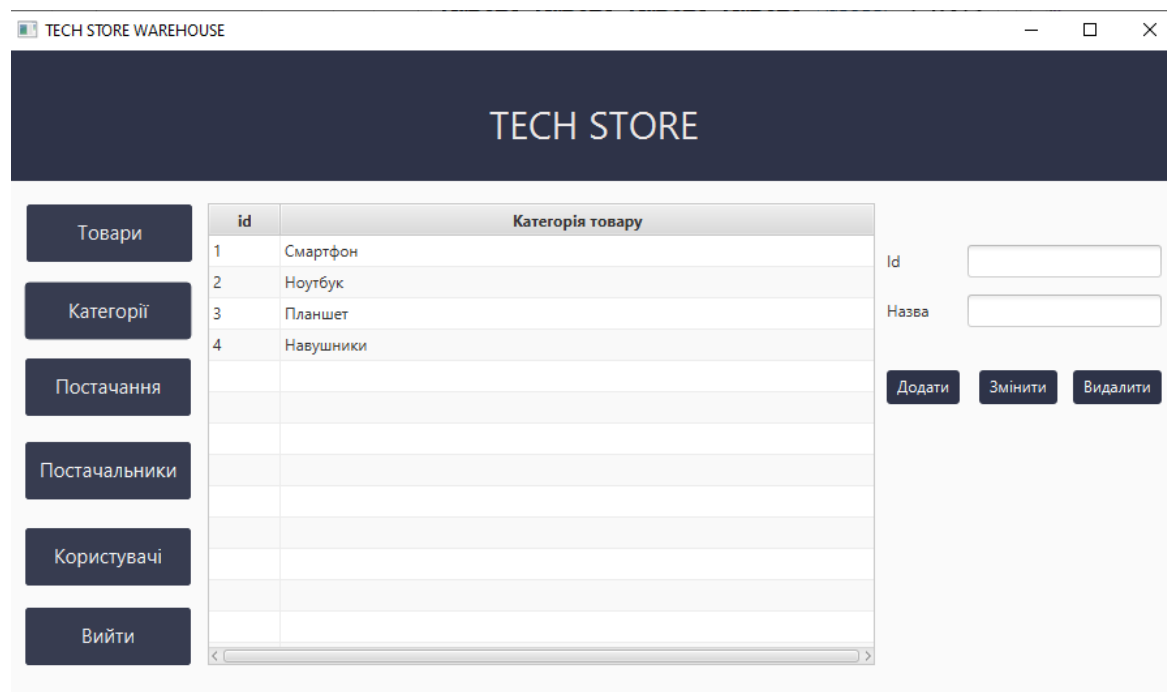
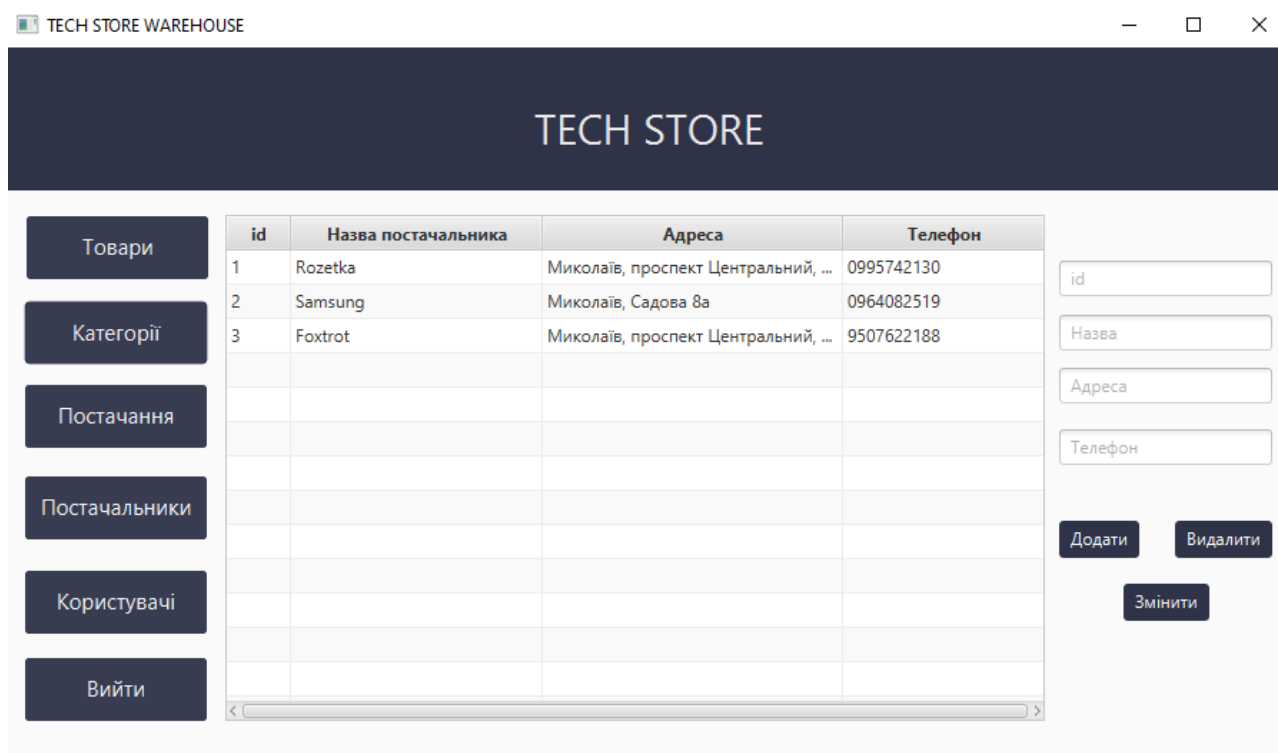


Рисунок Г4 – Вікно категорій товарів

Так само, як і у вікні з товарами, зліва є меню навігації, де можна перейти на інші сторінки програми.

5 – ВІКНО ПОСТАЧАЛЬНИКІВ

У вікні постачальників є таблиця зі списком постачальників, які привозять товари до складу магазину. Праворуч також є поля для додавання, видалення та редагування постачальника. Вікно постачальників показано на рисунку Г5.



id	Назва постачальника	Адреса	Телефон
1	Rozetka	Миколаїв, проспект Центральний, ...	0995742130
2	Samsung	Миколаїв, Садова 8а	0964082519
3	Foxtrot	Миколаїв, проспект Центральний, ...	9507622188

Рисунок Г5 – Вікно постачальників

Серед полів постачальника є код, назва постачальника, адреса, звідки товари приходять до складу та номер телефону.

6 – ВІКНО КОРИСТУВАЧІВ

Вікно користувачів представляє собою список користувачів, їх код, імена, пароль та номер телефону, а також можливість додавання, видалення та редагування. Вікно користувачів представлено на рисунку Г6.

id	Ім'я користувача	Пароль	Телефон
1	root	password	0996453423
2	vladislove	hehe	0665342812
3	rabotyaga	narabote	0934441352

Рисунок Г6 – Вікно користувачів

Робити операції з користувачами може тільки адміністратор програми.

ДОДАТОК Д – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1 ОБ'ЄКТ ВИПРОБУВАННЯ

Об'єктом випробування є програмне забезпечення для автоматизації управління складом, що розробляється на підставі завдання на кваліфікаційну роботу, виданого кафедрою ПЗАС НУК імені адмірала Макарова.

2 МЕТА ВИПРОБУВАННЯ

Метою проведення випробування є перевірка працездатності даного програмного забезпечення, перевірка відповідності характеристик та вимог розробленої програми, викладених у технічному завданні, приведення прикладу роботи та отримання відповідних навичок. При проведенні випробувань функціональні характеристики (можливості) програми підлягають перевірці на відповідність вимогам, викладеним у пункті «Вимоги до функціональних характеристик» технічного завдання.

3 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

Програмна документація повинна включати в себе наступні документи:

- Текст програми;

- Опис програми;
- Програма та методика випробувань;
- Інструкція користувача.

4 ЗАСОБИ І ПОРЯДОК ВИПРОБУВАНЬ

5.1 Технічні засоби, що використовуються під час випробувань

Склад технічних засобів, що використовувалися під час випробувань:

- Процесор: 2 ядра, 2.3 ГГц;
- Оперативна пам'ять: 8 ГБ;
- Вільного місця: 90 GB.

5.2 Програмні засоби, що використовуються під час випробувань

Склад програмних засобів, що використовувалися під час випробувань:

- Операційна система Windows 10 Pro;
- Java Runtime Environment.

5.3 Порядок проведення випробувань

Порядок проведення випробувань складається з перевірки вимог до функціональних характеристик програмного продукту та перевірки вимог до програмної документації.

6 МЕТОДИ ВИПРОБУВАНЬ

6.1 Методика проведення перевірки вимог до програмної документації

Перевірка дотримання вимог програмної документації на програмний продукт виконується візуально представником служби, відповідальної за експлуатацію. У ході перевірки зіставляється склад і комплектність програмної документації, представленої розробником, з переліком програмної документації, наведеним у пункті «Склад програмної документації, пропонованої на випробування» цього документа. Перевірка вважається завершеною у випадку відповідності складу та комплектності програмної документації, представленої розробником, переліком програмної документації, наведеному у зазначеному вище пункті.

6.2 Методика проведення перевірки вимог до функціональних характеристик програмного продукту

Перевірка працездатності програми виконується згідно з пунктом «Вимоги до функціональних характеристик» Додатку А Технічне завдання. Перевірка вважається завершеною у разі відповідності складу і послідовності виконаних дій пункту «Вимоги до функціональних характеристик». Була перевірена функціональність за темою «Розробка програмного забезпечення для автоматизації управління складом». У табл. Д.1, вказаний перелік випробувань основних функціональних можливостей.

Таблиця Д.1 – Методика випробування

№	Дія	Очікуваний результат	Результат перевірки	Зауваження
1	Перевірка правильності введення даних при авторизації	Програма повинна перевірити введені дані з записами у БД	Виконано	
2	Перевірка правильності додавання нового запису в БД	Програма повинна перевірити коректність введених даних	Виконано	
3	Перевірка правильності введених даних при редагуванні записів в БД	Програма повинна перевірити коректність введених даних	Виконано	
4	Перевірка правильності видалення запису з БД	Програма повинна перевірити введені дані з записами у БД	Виконано	