

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проєктами

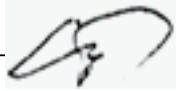
Кафедра програмного забезпечення автоматизованих систем

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»

Завідувач кафедри

_____ проф. Приходько С.Б.

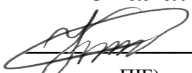
« ____ » _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «бакалавр»

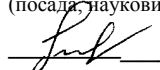
на тему: Розробка програми для автоматизації роботи з комітентами
дитячого комісійного магазину "Сонечко"

Виконала: студентка групи 4151з

 Ветлугіна К.О.
(підпис, ПІБ)

Керівник роботи:

доцент кафедри ПЗАС, к.т.н., доцент
(посада, науковий ступень вчене звання)

 Макарова Л. М.
(підпис, ПІБ)

Миколаїв – 2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

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

«ЗАТВЕРДЖУЮ»
 Гарант освітньої програми
_____ доц. Макарова Л.М.
« 08 » ____ 04 ____ 2025 р.

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

Студенту _____ Ветлугіній Крістіні Олександрівні
(Прізвище, ім'я, по батькові)

1. Тема роботи: Розробка програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

Керівник роботи _____ доцент кафедри ПЗАС, к.т.н., доцент Макарова Л. М
Затверджені наказом ректора №153 від 08.04.2025 р.

2. Термін подання роботи: 02.06.2025 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; Аналіз практичної задачі інженерії програмного забезпечення; Проєкт програмного забезпечення; 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 _____

Титульний слайд, Мета та завдання кваліфікаційної роботи, Аналіз вимог до програми, Архітектура програми, Діаграми діяльності програми, Концептуальна модель даних, Діаграма класів програми, Діаграми послідовності програми, Логічна модель даних програми, Результати розробки програми, Висновки, Заклучний слайд.

6. Консультанти розділів роботи

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

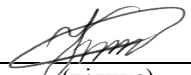
7. Дата видачі завдання 08.04.2025 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу ВСТУП	31.03.2025	ПП*
2.	Аналіз практичної задачі інженерії ПЗ	07.04.2025	ПП*
3.	Аналіз вимог до ПЗ	14.04.2025	ПП*
4.	Розробка архітектури ПЗ	21.04.2025	
5.	Розробка проєкту ПЗ	05.05.2025	
6.	Реалізація та тестування ПЗ	12.05.2025	
7.	Підготовка розділу Результати розробки ПЗ	16.05.2025	
8.	Підготовка розділу з охорони праці	19.05.2025	ПП*
9.	Підготовка розділу ВИСНОВКИ	23.05.2025	
10.	Оформлення списку використаних джерел та додатків	26.05.2025	
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.)	02.06.2025	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку
12.	Підготовка презентації та доповіді	06.06.2025	
13.	Попередній захист роботи на засіданні кафедри ПЗАС	09.06.2025	
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, а також Авторського договору з додатком	16.06.2025	

* - за результатами переддипломної практики (ПП), яка була з 03.02.2025 до 23.03.2025 р.

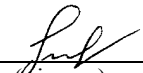
Студент


(підпис)

Встлугіна К.О.

(ПІБ)

Керівник роботи


(підпис)

Макарова Л.М.

(ПІБ)

АНОТАЦІЯ

Кваліфікаційна робота присвячена розв'язанню практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме розробці програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко", яка дозволить спростити та автоматизувати роботу з комітентами, зокрема полегшити ведення обліку розрахунків та формування відповідної звітності, зменшити кількість помилок.

В кваліфікаційній роботі представлені аналіз практичної задачі інженерії програмного забезпечення з теми розробки, аналіз існуючих програмних аналогів та постановка задачі на розробку програмного забезпечення. Розроблено архітектуру та проєкт програмного забезпечення. Представлено результати розробки та розділ з охорони праці.

Кваліфікаційна робота викладена на 80 сторінках машинописного тексту та містить: 14 рисунків, 26 таблиць, 5 додатків, список використаних джерел з 19 найменувань.

ABSTRACT

The qualification work is dedicated to solving the practical task of software engineering, characterized by complexity and uncertainty of conditions, namely Software development for automation of work with the committees of the children's commission store "Sunshine", which will simplify and automate the work with committees, in particular, facilitate the accounting of calculations and the formation of appropriate reporting, reduce the number of errors.

In qualification work presents analysis of the practical task of software engineering on the topic of development, analysis of existing software analogues and formulation of the problem for software development. Architecture and software project are developed. Results of development and the section on labor protection are presented.

The qualification work is presented on 80 pages of typewritten text and contains: 14 figures, 26 tables, 5 appendices, the list of references from 19 names.

ЗМІСТ

ВСТУП	7
1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ПРОГРАМИ ДЛЯ АВТОМАТИЗАЦІЇ РОБОТИ З КОМІТЕНТАМИ ДИТЯЧОГО КОМІСІЙНОГО МАГАЗИНУ "СОНЕЧКО"	9
1.1 Аналіз практичної задачі інженерії програмного забезпечення	9
1.2 Аналіз існуючих програмних продуктів	10
1.3 Постановка задачі на розробку програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"	13
2 РОЗРОБКА ПРОГРАМИ ДЛЯ АВТОМАТИЗАЦІЇ РОБОТИ З КОМІТЕНТАМИ ДИТЯЧОГО КОМІСІЙНОГО МАГАЗИНУ "СОНЕЧКО"	17
2.1 Аналіз вимог до програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"	17
2.1.1 Побудова моделі варіантів використання	17
2.1.2 Розробка специфікації варіантів використання	18
2.2 Архітектура програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"	24
2.3 Проект програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"	26
2.3.1 Ескізний проект програми	26
2.3.2 Технічний проект програми	32
2.3.3 Робочий проект програми	37
3 РЕЗУЛЬТАТИ РОЗРОБКИ ПРОГРАМИ ДЛЯ АВТОМАТИЗАЦІЇ РОБОТИ З КОМІТЕНТАМИ ДИТЯЧОГО КОМІСІЙНОГО МАГАЗИНУ "СОНЕЧКО"	43
4 ОХОРОНА ПРАЦІ	45

4.1 Загальні питання організації охорони праці	45
4.2 Вплив виробничого середовища при роботі з персональним комп'ютером	46
ВИСНОВКИ	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	52
Додаток А Технічне завдання на розробку програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"	54
Додаток Б Код програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"	60
Додаток В Опис програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"	71
Додаток Г Інструкція користувача програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"	74
Додаток Д Програма та методика випробування програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"	78

ВСТУП

У сучасних умовах стрімкого розвитку інформаційних технологій автоматизація бізнес-процесів є одним із ключових чинників підвищення ефективності діяльності підприємств. Особливо це актуально для малого бізнесу, зокрема комісійних магазинів, які займаються реалізацією товарів, наданих комітентами. Такі підприємства потребують зручних, надійних та доступних програмних рішень для обліку товарів, ведення бази комітентів, контролю реалізації й формування звітності. Зокрема, автоматизація дозволяє швидко обліковувати надходження, продаж і повернення, мінімізувати паперові записи і людські помилки, що є актуальним для формату комісійного обміну – адже товари належать не магазину, а їх власникам-комітентам [1, 2].

Комісійний магазин «Сонечко», який спеціалізується на реалізації дитячих товарів, стикається з необхідністю оптимізації внутрішніх процесів, зокрема роботи з комітентами. На даний момент ведення обліку здійснюється або вручну в паперовому вигляді, або за допомогою загальних офісних програм Microsoft Office, що призводить до втрати часу, помилок у розрахунках та труднощів у веденні звітності.

Виходячи з цього, можна сказати, що розробка програмного забезпечення для автоматизації роботи з комітентами є актуальною.

Метою кваліфікаційної роботи є розв'язання практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме розробка програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко", для чого потрібно, в першу чергу, виконати аналіз практичної задачі інженерії програмного забезпечення, зокрема, аналіз сучасних тенденцій автоматизації роботи з комітентами, а також доцільність розробки власного програмного забезпечення для автоматизації роботи дитячого комісійного магазину "Сонечко".

Комплексність зазначеної практичної задачі полягає у тому, що розробка

програми враховуватиме багато взаємопов'язаних між собою елементів, а також різноманітних вимог до функцій програми. Невизначеність умов зазначеної практичної задачі полягає у тому, що вимоги до програми можуть бути нечіткими або недостатньо визначеними на початковому етапі проектування.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- провести аналіз практичної задачі інженерії програмного забезпечення, яка вирішується у кваліфікаційній роботі;
- зробити висновок про доцільність розробки власної програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко";
- сформулювати вимоги до програми та розробити технічне завдання на розробку відповідної програми;
- обрати архітектурний стиль та розробити архітектуру програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко";
- розробити проєкт програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко", який включає ескізний, технічний та робочий проєкти цієї програми;
- провести тестування та випробування розробленої програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко";
- розробити усю необхідну документацію до програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко".

В результаті буде розроблено програму для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко", яка дозволить спростити та автоматизувати роботу з комітентами, зокрема полегшити ведення обліку розрахунків та формування відповідної звітності, зменшити кількість помилок.

Об'єктом кваліфікаційної роботи є процес розробки програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко".

1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ПРОГРАМИ ДЛЯ АВТОМАТИЗАЦІЇ РОБОТИ З КОМІТЕНТАМИ ДИТЯЧОГО КОМІСІЙНОГО МАГАЗИНУ "СОНЕЧКО"

1.1 Аналіз практичної задачі інженерії програмного забезпечення

Розробку програми для автоматизації роботи з комітентами дитячого комісійного магазину «Сонечко» слід розпочати з аналізу практичної задачі інженерії програмного забезпечення. На цьому етапі необхідно визначити коло бізнес-процесів, які потребують автоматизації, встановити ключові функціональні вимоги до майбутньої програми, а також оцінити типові проблеми, які виникають у поточній організації обліку.

Автоматизація бізнес-процесів є однією з центральних задач сучасної інженерії програмного забезпечення. Підприємства торговельної сфери, зокрема комісійні магазини, дедалі частіше потребують індивідуальних програмних рішень, які враховують їхні специфічні потреби – від обліку товарів, що належать третім особам (комітентам), до ведення фінансових звітів і управління клієнтською базою.

Сучасні підприємства рідко обходяться без програмного забезпечення для обліку, аналітики та управління. Найбільш поширеними є системи бухгалтерського, управлінського та складського обліку, CRM та інші спеціалізовані рішення. Проте, для малого бізнесу, зокрема, дитячих комісійних магазинів, використання дорогих комплексних систем часто є недоцільним. У таких випадках виникає потреба у створенні вузькоспеціалізованого, простого у використанні та недорогого програмного забезпечення.

На практиці облік роботи з комітентами у подібних магазинах ведеться вручну або за допомогою загальних офісних інструментів, що призводить до ряду проблем:

- підвищена ймовірність помилок при ручному введенні даних;
- значні часові витрати на повторне внесення однотипної інформації під час формування звітів;
- відсутність єдиного сховища даних і, як наслідок, складнощі у веденні аналітики;
- надмірне використання паперових документів.

В дитячому комісійному магазині «Сонечко» функції бухгалтера та адміністратора поєднує власник. Він самостійно веде облік, формує звітність за періодами (щодня, щотижня, щомісяця, поквартально) та веде облік взаєморозрахунків з кожним комітентом. Продавці ж відповідають за прийом товару, оформлення відповідних накладних та фіксацію фактів реалізації. З огляду на це, доцільно автоматизувати наступні аспекти діяльності:

- реєстрацію товарів, переданих на реалізацію;
- облік продажів по кожному товару та комітенту;
- автоматичне формування звітів за вказані періоди;
- облік виплат комітентам та нарахування комісії;
- зберігання історії операцій і друк необхідних документів.

Таким чином, розробка спеціалізованої програми дозволить значно спростити щоденну роботу персоналу магазину, мінімізувати кількість помилок, пришвидшити процес формування звітності та забезпечити більшу прозорість взаєморозрахунків з комітентами. Впровадження такої програми сприятиме підвищенню продуктивності праці, зниженню витрат і, відповідно, зростанню прибутку підприємства.

1.2 Аналіз існуючих програмних продуктів

Існує ряд програмних рішень для автоматизації торгівлі, але вони більш прийнятні для великих торговельних підприємств або торговельних мереж. Зазвичай, вони представляють собою комплексну систему автоматизації роботи

різних категорій працівників: керівників підприємств, менеджерів з продажу, бухгалтерів, і т.д. Інші, навпаки, вирішують більш конкретні задачі. Розглянемо деякі з них та проведемо аналіз можливості їх використання для автоматизації роботи з комітентами дитячого комісійного магазину «Сонечко».

Програми Мінісофт Комерція розроблені для різних галузей застосування та дозволяють автоматизувати у тому числі і роздрібні магазини [3]. Як заявлено виробником, вони мають досвід автоматизації магазинів різної спрямованості (взуття, одяг, дитячі товари, будівельні матеріали і сантехнічні вироби, косметика і побутова хімія) та різного розміру (маленькі або великі) [2].

Основні переваги програмного забезпечення:

- повний облік товарів і послуг (наявність і рух товарів, групування по категоріях, типам документів і звітам);
- повний облік готівкових і безготівкових коштів об'єкта, ведення розрахунків з постачальниками й операційних витрат;
- ведення бази клієнтів і гнучке формування системи їх лояльності;
- здійснення контролю дій персоналу.

Основні недоліки с точки зору застосування у дитячому комісійному магазині «Сонечко»:

- велика надлишкова функціональність (фінансовий та управлінський облік, система лояльності клієнтів, інструменти для обміну даними и т.д.);
- база для роботи з різними типами торгового обладнання, які не використовуються у роботі даного магазину.

Інше сімейство програм – Автоматизація магазину від розробника Skynum. Серед можливостей даних програмних продуктів відмічається:

- автоматизація обліку товару в магазині та ведення складського обліку;
- контроль терміну придатності товару;
- роздрібний та оптовий продаж;
- програма для касира магазину;

- CRM система;
- програма лояльності клієнтів;
- створення інтернет-магазину [4].

Основним недоліком з точки зору застосування у дитячому комісійному магазині «Сонечко» є те, що представлений сервіс є онлайн сервісом, тобто "хмарним", який потребує постійного сталого доступу до мережі інтернет.

Ще одна розглянута система для автоматизації та обліку торгової мережі від розробника PayKit – це каса, облік, ПРРО у POS-терміналі. Представляє собою мобільне робоче місце касира на телефоні або планшеті, яке не прив'язане до стаціонарної точки [5]. Компактне та просте у використанні, воно поєднує в собі усі можливості стаціонарно обладнаної точки продажу, але на сьогоднішній день такий сервіс не потрібен у дитячому комісійному магазині «Сонечко».

Перехід до універсальних хмарних рішень все більше стає представлений серед програмних продуктів для автоматизації торгівельної діяльності, тому наступне розглянуте рішення від розробника Ainur також представляє собою універсальну хмарну платформу, більше орієнтовану на роботу через мобільні додатки [6].

Останній програмний продукт, який було розглянуто у цьому огляді – Автоматизація магазину роздрібної торгівлі від розробника Торгсофт, який представляє собою класичну коробкову версію програми для встановлення безпосередньо на комп'ютері користувача.

Це універсальна програма обліку, яка підтримує:

- роздрібні та оптові магазини;
- CRM для інтернет-торгівлі та створення інтернет-магазинів;
- складський облік та облік виробництва;
- програма для сфери послуг, сервісних центрів та майстерень;
- облік оренди та прокату;

- облік у кафе;
- фінансовий та управлінський облік [7].

А, отже, основним недоліком с точки зору застосування у дитячому комісійному магазині «Сонечко», знов таки, є велика надлишкова функціональність даного програмного продукту.

Отже, з розглянутих існуючих програмних продуктів не було знайдено такого, який би повністю задовольнив поставленим вимогам для автоматизації роботи з комітентами дитячого комісійного магазину «Сонечко».

В результаті аналізу зроблено висновок про доцільність власної розробки – програми для автоматизації роботи з комітентами дитячого комісійного магазину «Сонечко», яке дозволить спростити та автоматизувати роботу з комітентами, зокрема полегшити ведення обліку розрахунків та формування відповідної звітності, зменшити кількість помилок.

1.3 Постановка задачі на розробку програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

Виходячи з попередньо проведеного аналізу практичної задачі інженерії програмного забезпечення, було прийнято рішення розробити власну програму для автоматизації роботи з комітентами дитячого комісійного магазину «Сонечко», призначенням якої є спрощення та автоматизація роботи з комітентами. У програми будуть користувачі одного типу з однаковою функціональністю.

Дане програмне забезпечення повинно вирішувати наступні функції:

- реєстрації та авторизації користувачів;
- зберігання та використання інформації про комітентів;
- зберігання та використання інформації про товар;
- зберігання та використання інформації з продажу товару;

– складання звітності.

Авторизація користувача

Авторизація користувача в програмному забезпеченні для надання функціональних можливостей. Обов'язковими полями є: логін та пароль.

Перегляд даних

Для перегляду даних потрібно обрати таблицю.

Пошук необхідної інформації

Функція пошуку необхідної інформації відбувається за певними параметрами: необхідно обрати таблицю та вести атрибути для пошуку.

Додавання нового запису

Для додавання нового запису потрібно обрати таблицю.

Редагування запису

Для редагування запису потрібно обрати таблицю та відповідний запис.

Видалення запису

Для видалення запису потрібно обрати таблицю та відповідний запис.

Формування документу про прийом товару від комітента

Формування накладної з переліком товарів, переданих на реалізацію до магазину.

Оформлення продажу товару

Фіксування факту продажу конкретного товару.

Формування звітів

Формування звітів повинно відбуватися автоматично після обрання необхідних відповідних параметрів.

Представлення даних

Данні представляються у вигляді наступних таблиць у БД: працівник, комітент, товар, продажі. Атрибути таблиць та їх опис наведено у таблицях 1.1 – 1.5.

Таблиця 1.1 – Опис атрибутів таблиці БД «Працівник»

Атрибут	Тип даних
Код працівника	Число
ПІБ працівника	Текст
Адреса	Текст
Телефон	Текст
Дата прийняття на роботу	Дата
Дата звільнення з роботи	Дата

Таблиця 1.2 – Опис атрибутів таблиці БД «Комітент»

Атрибут	Тип даних
Код комітента	Число
ПІБ комітента	Текст
Адреса	Текст
Телефон	Текст

Таблиця 1.3 – Опис атрибутів таблиці БД «Товар»

Атрибут	Тип даних
Код товару	Число
Тип товару	Текст
Найменування товару	Текст

Таблиця 1.4 – Опис атрибутів таблиці БД «Прийом товару»

Атрибут	Тип даних
Код прийому	Число
Код комітента	Число
Код товару	Число
Кількість	Число
Ціна продажу	Число

Таблиця 1.5 – Опис атрибутів таблиці БД «Продажі»

Атрибут	Тип даних
Код продажі	Число
Код працівника	Число
Код комітента	Число
Код прийому	Число
Кількість	Число
Дата продажу	Дата

Було виділено вимоги до організаційних вхідних та вихідних даних. Дані зберігаються в файлі БД з довільним доступом типу .myd.

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 та принтер).

Було виділено вимоги до складу та параметрів технічних засобів. Система повинна задовольняти наступним вимогам в мінімальній комплектації:

- CPU: Intel(R) i5 2,16 МГц або аналогічний за параметрами AMD;
- RAM: 4 GB;
- HDD: 10 GB вільного місця.

Також є вимоги до інформаційної та програмної сумісності. Для повноцінного функціонування системи необхідно використовувати операційну систему Windows версії не нижче 7. Необхідна наявність встановленого Standard Edition Development Kit не нижче 17.

Детальна постановка задачі на розробку програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" представлена у технічному завданні, яке наведено у додатку А.

2 РОЗРОБКА ПРОГРАМИ ДЛЯ АВТОМАТИЗАЦІЇ РОБОТИ З КОМІТЕНТАМИ ДИТЯЧОГО КОМІСІЙНОГО МАГАЗИНУ "СОНЕЧКО"

2.1 Аналіз вимог до програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

2.1.1 Побудова моделі варіантів використання

UML – уніфікована мова моделювання, яка призначена для візуалізації та специфікації, конструювання та документування будь-яких програмних систем: розподілених веб-додатків, вбудованих систем, систем реального часу та інших. Використовуючи засоби UML, можна описати програмну систему з різних точок зору: проектування, розробка, розгортання програмного забезпечення.

Діаграма варіантів використання – це UML-діаграма, за допомогою якої в графічному вигляді можна представити вимоги до програмного забезпечення, що розробляється. Діаграма варіантів використання – це перша концептуальна модель програмного забезпечення, що розробляється. Варіант використання відображає сервіси, які програмне забезпечення надає користувачу [8].

З програмою для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко", що розробляється, буде працювати одна особа – Користувач, який матиме доступ до всього функціоналу програми.

Діаграма варіантів використання програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" наведена на рис. 2.1.



Рисунок 2.1 – Діаграма варіантів використання програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

Далі потрібно розробити специфікації кожного варіанту використання програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко".

2.1.2 Розробка специфікації варіантів використання

Розглянемо більш детально кожний варіант використання програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко". Потік подій кожного варіанту використання складається з:

- короткого опису;

- передумов;
- основного потоку подій;
- альтернативного потоку подій;
- постумов.

Детальний опис цих складових частин для кожного варіанту використання програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" наведений у таблицях 2.1 – 2.8.

Таблиця 2.1 – Опис варіанту використання "Авторизація"

Складова	Опис
Короткий опис	Дана функція дозволяє користувачу пройти перевірку для доступу до системи.
Передумови	Система запущена та виводить форму для авторизації.
Основний потік подій	1. Система виводить форму для вводу логіну та паролю. 2. Користувач вводить логін та пароль і підтверджує ввід. 3. Система перевіряє вірність логіну та паролю. 4. Якщо дані достовірні, то завантажується головне вікно програми.
Альтернативний потік подій	1. Система виводить форму для вводу логіну та паролю. 2. Користувач вводить логін та пароль і підтверджує ввід. 3. Система перевіряє вірність логіну та паролю. 4. Якщо дані недостовірні, повідомлення про те, що логін або пароль не вірні. 5. Надання можливості повторного вводу логіну та паролю.
Постумови	Користувач авторизований в системі.

Таблиця 2.2 – Опис варіанту використання "Пошук інформації"

Складова	Опис
1	2
Короткий опис	Дана функція дозволяє користувачу здійснювати пошук необхідної інформації.

Продовження таблиці 2.2

1	2
Передумови	Користувач авторизований у системі.
Основний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмою. 3. Користувач обирає таблицю по якій буде відбуватись пошук необхідної інформації. 4. Користувач вводить необхідні дані для пошуку та натискає кнопку «Знайти».
Альтернативний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмою. 3. Користувач обирає таблицю по якій буде відбуватись пошук необхідної інформації. 4. Користувач вводить необхідні дані для пошуку та натискає кнопку «Відміна».
Постумови	Вивід знайденої інформації.

Таблиця 2.3 – Опис варіанту використання "Додавання запису"

Складова	Опис
Короткий опис	Дана функція дозволяє користувачу додати новий запис до таблиці
Передумови	Користувач авторизований у системі.
Основний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмою. 3. Користувач обирає та відкриває таблицю. 4. Для додавання нового запису у таблицю, користувач натискає кнопку «Додати». З'являється відповідна панель, де вносяться дані. 5. Збереження внесених даних за допомогою натискання на кнопку «Зберегти».
Альтернативний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмою. 3. Користувач обирає та відкриває таблицю. 4. Для додавання нового запису у таблицю, користувач натискає кнопку «Додати». З'являється відповідна панель, де вносяться дані. 5. Для відміни запису даних, користувач натискає кнопку «Відміна».
Постумови	Додано новий запис до таблиці.

Таблиця 2.4 – Опис варіанту використання "Редагування запису"

Складова	Опис
Короткий опис	Дана функція дозволяє користувачу відредагувати вже існуючий запис у таблиці.
Передумови	Користувач авторизований у системі.
Основний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмною. 3. Користувач обирає та відкриває таблицю. 4. Далі робота відбувається на панелі «Дані таблиці». 5. Для редагування вже існуючого запису, користувач обирає запис та натискає кнопку «Редагувати». З'являється відповідна панель, де редагуються дані. 6. Збереження внесених змін за допомогою натискання на кнопку «Зберегти».
Альтернативний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмною. 3. Користувач обирає та відкриває таблицю. 4. Далі робота відбувається на панелі «Дані таблиці». 5. Для редагування вже існуючого запису, користувач обирає запис та натискає кнопку «Редагувати». З'являється відповідна панель, де редагуються дані. 6. Для відміни запису даних, користувач натискає кнопку «Відміна».
Постумови	Збереження та вивід відредагованого запису.

Таблиця 2.5 – Опис варіанту використання "Видалення запису"

Складова	Опис
1	2
Короткий опис	Дана функція дозволяє користувачу видалити вже існуючий запис у таблиці.
Передумови	Користувач авторизований у системі.

Продовження таблиці 2.5

1	2
Основний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмною. 3. Користувач обирає та відкриває таблицю. 4. Натискає кнопку «Видалити» 5. Підтверджує внесені зміни за допомогою натискання на кнопку «Зберегти».
Альтернативний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмою. 3. Користувач обирає та відкриває таблицю. 4. Натискає кнопку «Видалити» 5. Для відміни видалення запису, користувач натискає кнопку «Відміна».
Постумови	Запис видалено.

Таблиця 2.6 – Опис варіанту використання "Облік прийому товару від комітента"

Складова	Опис
Короткий опис	Дана функція дозволяє користувачу здійснити облік прийому товару від комітента.
Передумови	Користувач авторизований у системі.
Основний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмною. 3. Користувач обирає таблицю «Прийом товару» та натискає кнопку «Додати». 4. З'являється відповідне вікно, де користувач заповнює усі необхідні поля і натискає кнопку «Зберегти».
Альтернативний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмною. 3. Користувач обирає таблицю «Прийом товару» та натискає кнопку «Додати». 4. З'являється відповідне вікно, де користувач заповнює усі необхідні поля і натискає кнопку «Відміна».
Постумови	Оформлено новий прийом товару від комітента.

Таблиця 2.7 – Опис варіанту використання "Облік продажу товару"

Складова	Опис
Короткий опис	Дана функція дозволяє користувачу оформити продаж товару.
Передумови	Користувач авторизований у системі.
Основний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмою. 3. Користувач обирає таблицю «Продаж» та натискає кнопку «Додати». 4. З'являється відповідне вікно, де користувач обирає товар та інші дані для формування продажу і натискає кнопку «Сформувати». 5. З'являється сформована продаж товару, за необхідності користувач натискає кнопку «Друк».
Альтернативний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмною. 3. Користувач обирає таблицю «Продаж» та натискає кнопку «Додати». 4. З'являється відповідне вікно, де користувач обирає товар та інші дані для формування продажу і натискає кнопку «Відміна».
Постумови	Оформлено новий продаж товару.

Таблиця 2.8 – Опис варіанту використання "Формування звітів"

Складова	Опис
1	2
Короткий опис	Дана функція дозволяє користувачу сформувати необхідний звіт.
Передумови	Користувач авторизований у системі.
Основний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмною. 3. Користувач обирає дію «Звіт». 4. З'являється відповідне вікно, де користувач обирає тип звіту та всі необхідні дані для цього типу звіту і натискає кнопку «Сформувати». 5. З'являється сформований звіт, користувач натискає кнопку «Друк».

Продовження таблиці 2.8

1	2
Альтернативний потік подій	1. Авторизація. 2. Система виводить головне вікно роботи з програмною. 3. Користувач обирає дію «Звіт». 4. З'являється відповідне вікно, де користувач обирає тип звіту та всі необхідні дані для цього типу звіту і натискає кнопку «Відміна».
Постумови	Сформовано для друку новий звіт

Таким чином, використовуючи детальні специфікації варіантів використання програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко", можемо приступати до подальшого проектування вказаного програмного забезпечення.

2.2 Архітектура програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

Архітектура програми – це її базова організація, реалізована через компоненти програмного забезпечення та їх відносини. Вона демонструє принципи, що покладені при проектуванні програмного забезпечення, а також напрямок майбутнього розвитку цього програмного забезпечення [9].

Архітектура програмного забезпечення складається з усіх прийнятих проектних рішень з приводу структури програмного забезпечення, а також взаємодій між компонентами структури. Ці рішення забезпечать потрібний набір властивостей, які повинно підтримувати програмного забезпечення, щоб бути успішним. Проектні рішення надають концептуальну основу для розробки програмного забезпечення, його підтримки та обслуговування.

Архітектурним стилем програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" було обрано архітектуру, побудовану

навкруги бази даних, тому що:

- однократне введення та багатократне використання інформації є типовим при використанні бази даних;
- бази даних зазвичай будуються незалежно від прикладних програм, що розробляються з їх використанням;
- для бази даних є характерною структурованістю, яка відображає конкретну предметну область;
- бази даних зазвичай не дублюють одні і ті ж самі дані при їх використанні різними користувачами;
- бази даних забезпечують дотримання стандартів подання даних, що, в свою чергу, спрощує їх обслуговування;
- бази даних забезпечують централізоване керування даними та підтримку цілісності даних.

Архітектурний стиль програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" представлено на рисунку 2.2.

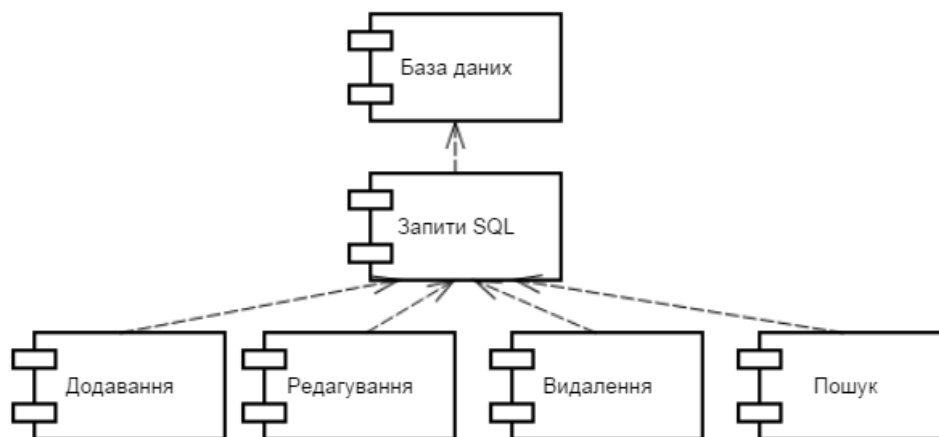


Рисунок 2.2 – Архітектура програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

База даних у даній архітектурі – централізоване сховище даних про комітентів дитячого комісійного магазину "Сонечко" та його працівників, які мають доступ до роботи програмного забезпечення, у якому за допомогою SQL-запитів здійснюється додавання (занесення нових даних), редагування (здійснення

оновлення даних, які не відповідають дійсності), видалення та пошук (за певним критерієм) відповідної інформації .

SQL-запити слугують засобом для взаємодії користувача з базою даних, що застосовується для формування запитів, оновлення (редагування) і керування (додавання, видалення, пошук) базою даних.

2.3 Проєкт програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

2.3.1 Ескізний проєкт програми

Сценарії основних варіантів використання

Кожен об'єкт програмного забезпечення має певну поведінку та може або знаходитися в якомусь стані, або переходити з одного стану в інший. В процесі такого переходу він здійснює певні дії, або виконує певний сценарій поведінки. Можна сказати, що поведінка об'єкта відображається його станами. Тому для відображення поведінки об'єкта можна використовувати один з двох видів діаграм: або діаграму станів, або діаграму активності [10].

Сценарії декількох варіантів використання програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" у вигляді діаграми діяльності наведено на рисунках 2.3 – 2.5.

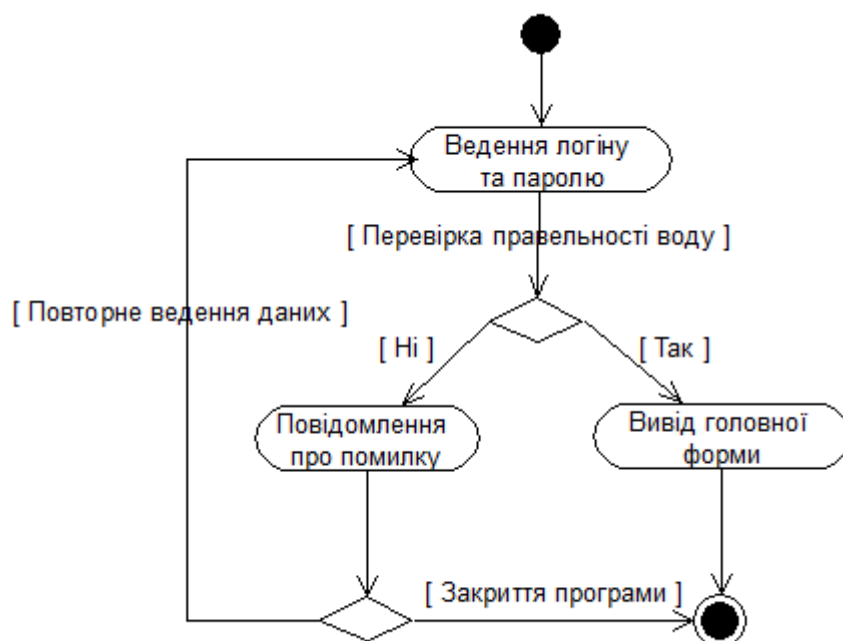


Рисунок 2.3 – Діаграма діяльності для варіанту використання «Авторизація» програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

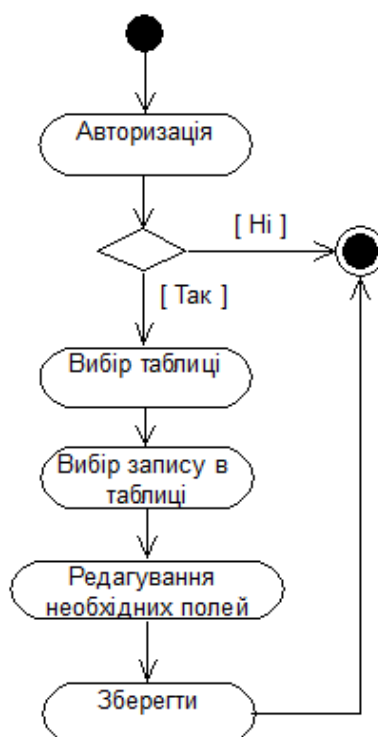


Рисунок 2.4 – Діаграма діяльності для варіанту використання «Редагування запису» програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

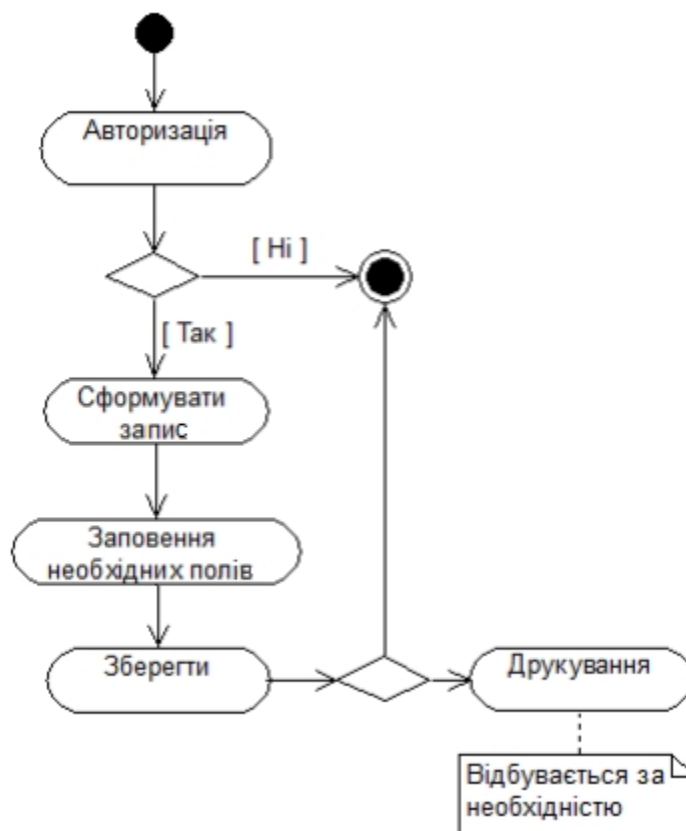


Рисунок 2.5 – Діаграма діяльності для варіанту використання «Облік прийому товару від комітента» програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

Розробка концептуальної моделі даних

На основі аналізу предметної галузі та постановки задачі було розроблено концептуальну модель даних програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" – модель, яка відображає структуру програмного забезпечення, властивості його елементів та зв'язки між ними, притаманні програмі та суттєві для розуміння основної мети.

Модель «сутність – зв'язок» є концептуальною моделлю, яка не враховує особливості конкретної системи управління базами даних. Найбільше поширення модель «сутність – зв'язок» отримала у вигляді так званої ER – моделі (Entity – relationship model), запропонованої Пітером Ченом[11].

Концептуальна даних програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" наведена на рис. 2.6.

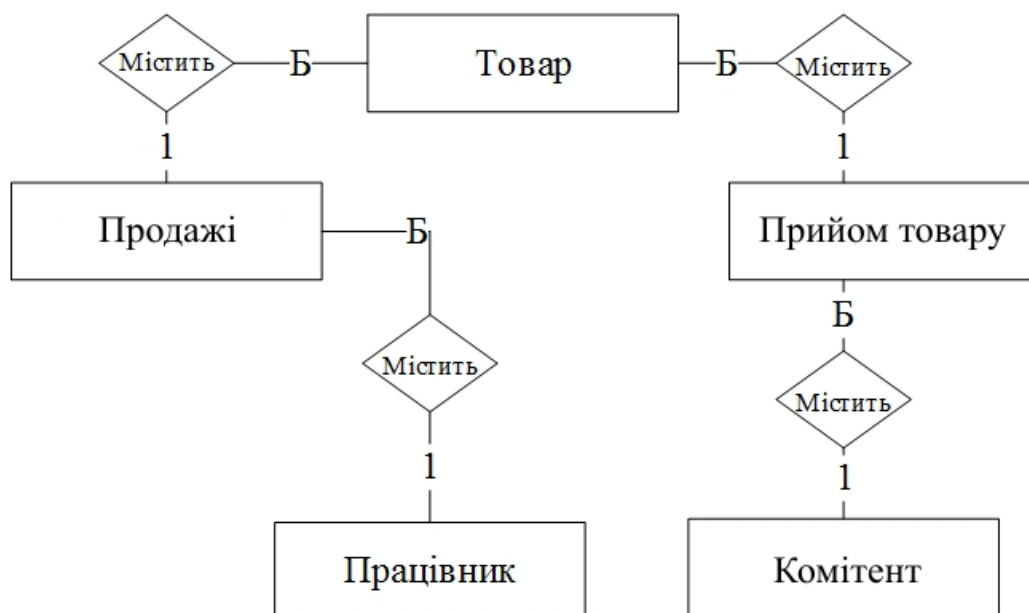


Рисунок 2.6 – Концептуальна модель даних програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

Розглянемо більш детально концептуальну модель даних програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" та виділимо атрибути кожної сутності.

Сутність «Працівник» має наступні атрибути:

- Код працівника;
- ПІБ працівника;
- Адреса;
- Телефон;
- Дата прийняття на роботу;
- Дата звільнення з роботи.

Сутність «Комітент» має наступні атрибути:

- Код комітента;
- ПІБ комітента;
- Адреса;
- Телефон.

Сутність «Товар» має наступні атрибути:

- Код товару;
- Тип товару;
- Найменування товару;

Сутність «Прийом товару» має наступні атрибути:

- Код прийому;
- Код комітента;
- Код товару;
- Кількість;
- Ціна продажу.

Сутність «Продажі» має наступні атрибути:

- Код продажі;
- Код працівника;
- Код комітента;
- Код прийому;
- Кількість;
- Дата продажу.

Ескізи інтерфейсу програмного забезпечення

Виходячи з опису діаграми варіантів використання програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко", було розроблено інтерфейс користувача даної програми. Інтерфейс користувача – це компоненти та елементи вихідних форм програмного забезпечення, які впливають на ефективну взаємодію між користувачем та програмним забезпеченням [12]. Розроблений ескіз графічного інтерфейсу програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" наведено на рис. 2.7 – 2.8.

Авторизація користувач

Логін

Пароль

Рисунок 2.7 – Ескіз форми авторизації програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

Головна форма

Товар
Працівник
Комітент
Прийом товару
Продажі

Пошук

Рисунок 2.8 – Ескіз головної форми програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

Маючи розроблений ескізний проєкт програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко", можемо переходити до наступної стадії проєктування – розробки технічного проєкту програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко".

2.3.2 Технічний проєкт програми

Діаграма класів в об'єктно-орієнтованому моделюванні призначена для представлення статичної структури програмного забезпечення в термінології класів об'єктно-орієнтованого програмування. Вона може відображати різні аспекти взаємозв'язків між різними сутностями предметної галузі, зокрема, об'єктами і підсистемами. Також вона призначена для описує їх внутрішньої структури та відносин між ними [13]. Діаграма класів представляє собою подальший розвиток концептуальної моделі проєктованої програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" та наведена на рис. 2.9.

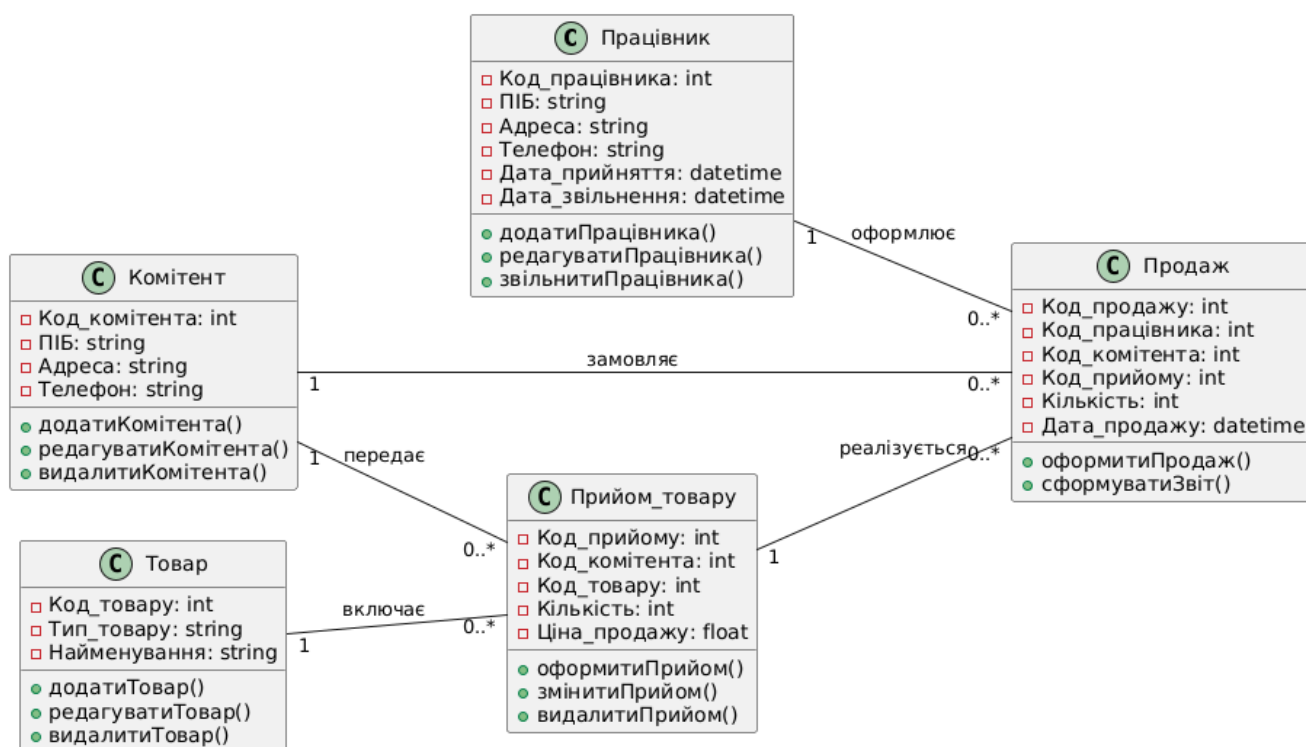


Рисунок 2.9 – Діаграма класів програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

Розробимо більш детальну специфікацію класів програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко", яка наведена в табл. 2.9 – 2.13.

Таблиця 2.9 – Специфікація класу "Працівник"

Складова	Опис
Відповідальність	Даний клас відповідає за інформацію по працівникам
Атрибут	Код працівника, ПІБ працівника, адреса, телефон, дата прийняття на роботу, дата звільнення з роботи
Операції зі специфікаціями операцій	додати Працівника() редагувати Працівника() звільнити Працівника()

Таблиця 2.10 – Специфікація класу "Комітент"

Складова	Опис
Відповідальність	Даний клас відповідає за інформацію по комітентам
Атрибут	Код комітента, ПІБ комітента, адреса, телефон
Операції зі специфікаціями операцій	додати Комітента() редагувати Комітента () видалити Комітента ()

Таблиця 2.11 – Специфікація класу "Товар"

Складова	Опис
Відповідальність	Даний клас відповідає за інформацію по товарам
Атрибут	Код товару, тип товару, найменування товару
Операції зі специфікаціями операцій	додати Товар () редагувати Товар () видалити Товар ()

Таблиця 2.12 – Специфікація класу "Прийом товару"

Складова	Опис
Відповідальність	Даний клас відповідає за інформацію по прийому товарів від комітентів
Атрибут	Код прийому, код комітента, код товару, кількість, ціна продажу
Операції зі специфікаціями операцій	оформити Прийом () змінити Прийом () видалити Прийом ()

Таблиця 2.13 – Специфікація класу "Продажі"

Складова	Опис
Відповідальність	Даний клас відповідає за інформацію по продажу товарів
Атрибут	Код продажу, код працівника, код комітента, код прийому, кількість, дата продажу
Операції зі специфікаціями операцій	оформити Продаж() сформувати Звіт()

Після розробки детальної специфікації класів програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" можна побудувати динамічну модель програми.

Динамічна модель програми

Динамічну модель програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" можна представити у вигляді основних діаграм послідовностей. Діаграми послідовностей можна віднести до діаграм взаємодії, які представляють поведінкові аспекти програмного забезпечення та розглядають взаємодію об'єктів у часі, тобто відображають часові особливості роботи об'єктів. Діаграми послідовностей можна використовувати для уточнення та доповнення діаграм прецедентів та більш детального опису логіки роботи сценаріїв використання [14]. Динамічна модель програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" у вигляді деяких діаграм послідовностей наведена на рис. 2.10 – 2.12.

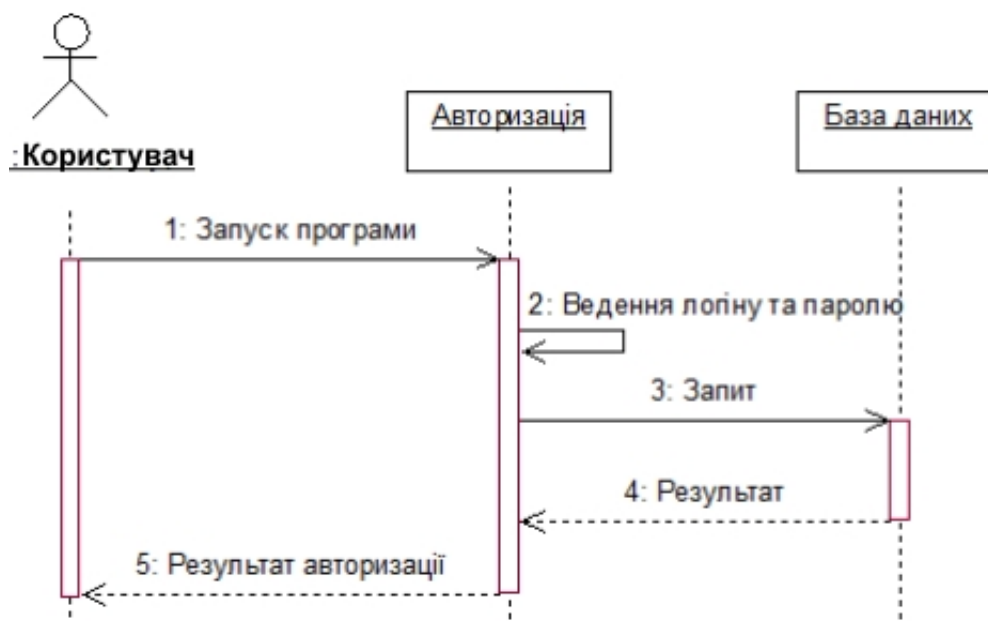


Рисунок 2.10 – Діаграма послідовності для прецеденту «Авторизація» програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

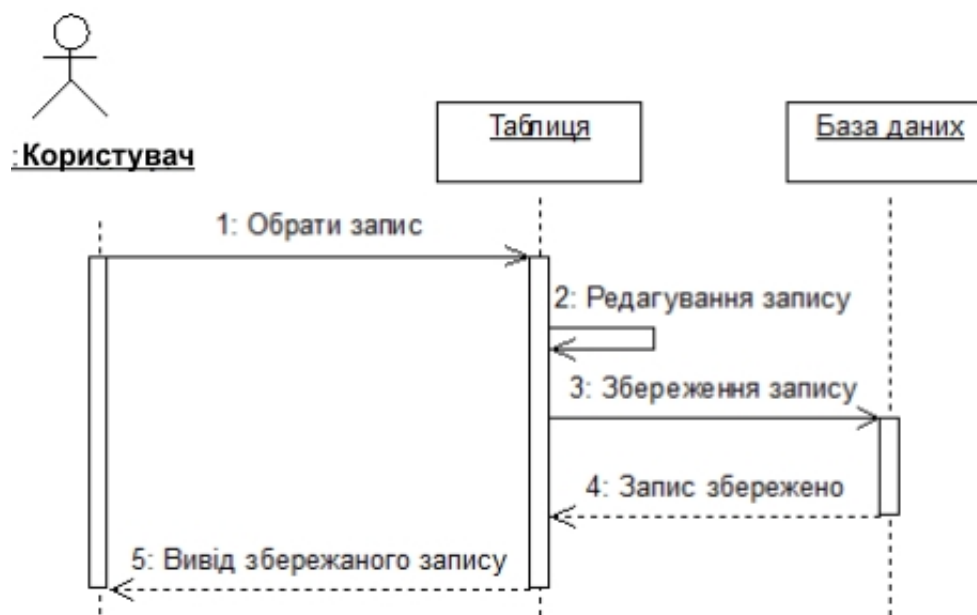


Рисунок 2.11 – Діаграма послідовності для прецеденту «Редагування запису» програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

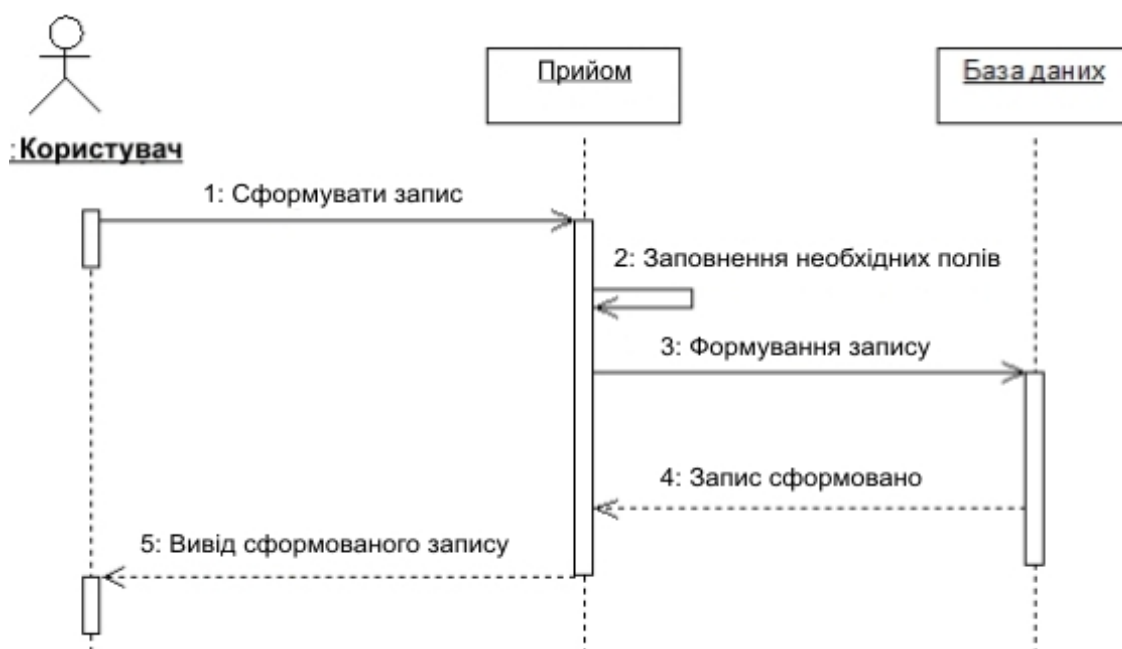


Рисунок 2.12 – Діаграма послідовності для прецеденту «Облік прийому товару від комітента» програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

Розробка логічної моделі бази даних

Для розробленої концептуальної моделі програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" необхідно представити предметну область у вигляді логічної моделі даних, яка може бути використана для розробки програми незалежно від вибору мови програмування та середовища розробки програми.

На основі концептуальної моделі програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко", наведеної на рис. 2.6, було розроблено логічну модель бази даних. Логічна модель бази даних розробляється в термінах інформаційних одиниць, але вона не прив'язана до якоїсь конкретної системи управління базами даних. Логічна модель бази даних є першим кроком до побудови фізичної моделі бази даних. Логічна модель бази даних програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" наведена на рис. 2.13.

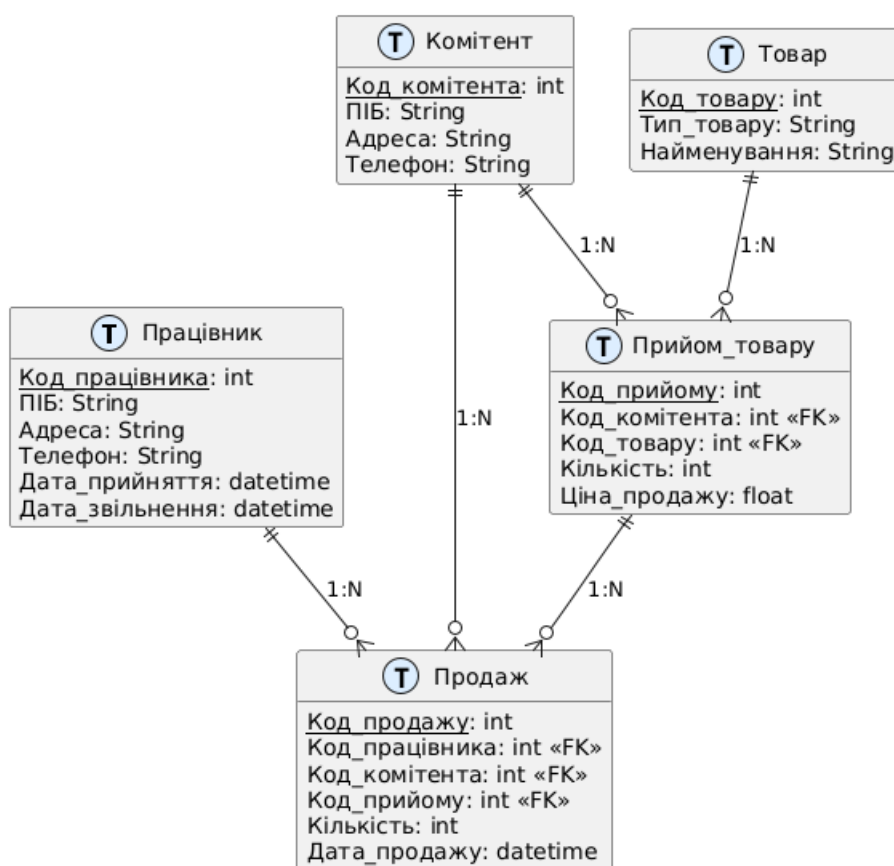


Рисунок 2.13 – Логічна модель бази даних програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

Маючи розроблений технічний проєкт програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко", можемо переходити до наступної стадії проєктування – розробки робочого проєкту програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко".

2.3.3 Робочий проєкт програми

Обґрунтування засобів розробки програмного забезпечення

У процесі розробки програмного забезпечення вибір мови програмування часто має визначальний вплив на ефективність створення, супроводу та масштабування майбутньої системи. Деякі мови є універсальними й дають змогу вирішувати задачі будь-якого рівня складності, інші – спеціалізовані та

орієнтовані на вузьке коло застосувань. Важливо, щоб синтаксис і семантика обраної мови були зручними для розробника, забезпечували зрозумілість коду, зменшували кількість помилок і сприяли підвищенню продуктивності праці.

Після аналізу практичної задачі інженерії програмного забезпечення, дослідження наявних підходів до вирішення поставленої задачі, вивчення технічних вимог до програмного забезпечення (див. Додаток А), а також порівняння існуючих технологічних рішень, були обрані оптимальні інструменти для реалізації програмного продукту, зокрема мова програмування, система управління базами даних (СУБД) та середовище розробки.

Основною мовою реалізації було обрано Java – об'єктно-орієнтовану мову програмування, яка з моменту своєї появи у 1995 році (розроблена компанією Sun Microsystems) отримала широке визнання у професійному середовищі. З 2009 року подальшим розвитком мови займається корпорація Oracle, яка підтримує її відповідно до стандартів Java Community Process. Java має синтаксис, подібний до мов C/C++, що полегшує її вивчення розробниками з відповідним досвідом. Програми на Java компілюються у проміжний байт-код, який виконується на будь-якій платформі за допомогою Java Virtual Machine (JVM). Це забезпечує високий рівень кросплатформеності. Засоби компіляції та віртуальна машина Java поширюються на умовах ліцензії GNU General Public License (GPL) [15, 16].

Для організації збереження та обробки даних у проєкті використовується MySQL – одна з найпоширеніших реляційних систем управління базами даних з відкритим вихідним кодом. MySQL підтримує використання як у невеликих локальних додатках, так і у веборієнтованих системах середнього рівня складності. Вона може функціонувати як мережевий сервер або вбудовуватись у автономні програми завдяки підтримці бібліотеки вбудованого сервера, що входить до стандартного дистрибутива. MySQL розповсюджується на умовах GPL або комерційної ліцензії [17].

Як середовище для розробки обрано JetBrains IntelliJ IDEA – одну з найбільш функціонально насичених та популярних інтегрованих середовищ розробки (IDE) для мови Java. IntelliJ IDEA надає широкий спектр можливостей:

інтелектуальний редактор коду, інструменти для автоматизованого рефакторингу, візуальний редактор графічного інтерфейсу, підтримку систем контролю версій, інтеграцію з фреймворками тестування (Ant, JUnit), засоби статичного аналізу коду та інші компоненти, що значно спрощують процес розробки. Вказані функціональні можливості дозволяють зменшити обсяг рутинної роботи, швидко виявляти й усувати помилки, а також підвищити загальну якість програмного продукту [18].

Розробка фізичної моделі бази даних

На основі логічної моделі бази даних було побудовано фізичну модель бази даних програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко", яка наведена в таблицях 2.14 – 2.18.

Таблиця 2.14 – Опис структури таблиці «Employee»

Ім'я поля	Тип поля	Ключ
employee_id	int	PK
full_name	varchar(255) NOT NULL	
address	varchar(255)	
phone	varchar(20)	
hire_date	date	
dismissal_date	date	

Таблиця 2.15 – Опис структури таблиці «Consignor»

Ім'я поля	Тип поля	Ключ
consignor_id	int	PK
full_name	varchar(255) NOT NULL	
address	varchar(255)	
phone	varchar(20)	

Таблиця 2.16 – Опис структури таблиці «Product»

Ім'я поля	Тип поля	Ключ
product_id	int	PK
product_type	varchar(100)	
product_name	varchar(255)	

Таблиця 2.17 – Опис структури таблиці «ProductReceipt»

Ім'я поля	Тип поля	Ключ
receipt_id	int	PK
consignor_id	int	FK
product_id	int	FK
quantity	int NOT NULL	
selling_price	decimal(10, 2)	

Таблиця 2.18 – Опис структури таблиці «Sale»

Ім'я поля	Тип поля	Ключ
sale_id	int	PK
employee_id	int	FK
consignor_id	int	FK
receipt_id	int	FK
quantity	int NOT NULL	
sale_date	date	

Реалізація та тестування основних класів еквівалентності програмного забезпечення

При тестуванні програма для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" розглядалась як чорний ящик. Тестування зводилися до послідовного вводу тестових наборів даних і аналізу отриманих після цього результатів.

Проаналізуємо правильні та неправильні класи еквівалентності, проведемо тестування на правильність роботи програма для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" в різних ситуаціях. Реалізація класів проводиться на основі діаграми класів програма для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко". Для прикладу наведені тести до класу «Комітент».

В таблиці 2.19 представлені класи еквівалентності та граничні значення вхідних даних:

- код комітента;
- ПІБ комітента;
- адреса;

– телефон.

Таблиця 2.19 – Класи еквівалентності вхідних даних для класу «Комітент»

Вхідні умови	Правильні класи еквівалентності	Неправильні класи еквівалентності
Код комітента	Число	Символи, які відрізняються від цифр
ПІБ комітента	Рядок	Графічні зображення (в тому числі і смайлики)
Адреса	Рядок	Графічні зображення (в тому числі і смайлики)
Телефон	Рядок	Графічні зображення (в тому числі і смайлики)

Почнемо з тестування для правильних класів еквівалентності класу «Комітент», яке наведено в табл.2.20. За результатами правильних класів еквівалентності програма для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" працює правильно.

Таблиця 2.20 – Тести з правильних класів еквівалентності для класу «Комітент»

№ тесту	Вхідні дані	Вихідні дані
1	В поле «Код комітента» введено число 19.	Виведення на екран кода комітента. Результат вірний.
2	В поле «ПІБ комітента» введено Гриценко Олена Петрівна.	Виведення на екран ПІБ комітента. Результат вірний.
3	В поле «Адреса» введено адресу комітента – вул. Дачна, 7, кв. 7.	Виведення на екран адреси комітента. Результат вірний.
4	В поле «Телефон» введено телефон працівника – 099-98-76-543.	Виведення на екран телефону комітента. Результат вірний.

Протестуємо неправильні класи еквівалентності класу «Комітент», що наведено в табл. 2.21.

Таблиця 2.21 – Тести з неправильних класів еквівалентності класу «Комітент»

№ тесту	Вхідні дані	Вихідні дані
1	2	3
1	В поле «Код комітента» введемо символи "йцуке"	Програма виводить повідомлення про помилку. Результат підтверджено
2	В поле «ПІБ комітента» ведемо графічний елемент – смайлик	Після ведення графічного елемента смайлика програма виводиться повідомлення про помилку. Результат підтверджено
3	В поле «Адреса» ведемо будь-який графічний символ ASCII.	Після ведення графічного символу ASCII програма виводиться повідомлення про помилку. Результат підтверджено
4	В поле «Телефон» ведемо графічний елемент – смайлик	Після ведення графічного елемента смайлика програма виводиться повідомлення про помилку. Результат підтверджено

За результатами тестування неправильних класів еквівалентності класу «Комітент», програма для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" працює правильно, не допускаючи до вводу ніякі елементи, окрім стандартних символів вводу та випадального списку.

3 РЕЗУЛЬТАТИ РОЗРОБКИ ПРОГРАМИ ДЛЯ АВТОМАТИЗАЦІЇ РОБОТИ З КОМІТЕНТАМИ ДИТЯЧОГО КОМІСІЙНОГО МАГАЗИНУ "СОНЕЧКО"

В даній кваліфікаційній роботі була розв’язана практична задача інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов – була розроблена програма для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко". Розроблена програма дозволить спростити та автоматизувати роботу з комітентами, зокрема полегшити ведення обліку розрахунків та формування відповідної звітності, та зменшити кількість помилок. Розробка програми виконана згідно технічного завдання, наведеного у Додатку А. Зовнішній вигляд головного вікна програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" наведено на рисунку 3.1.

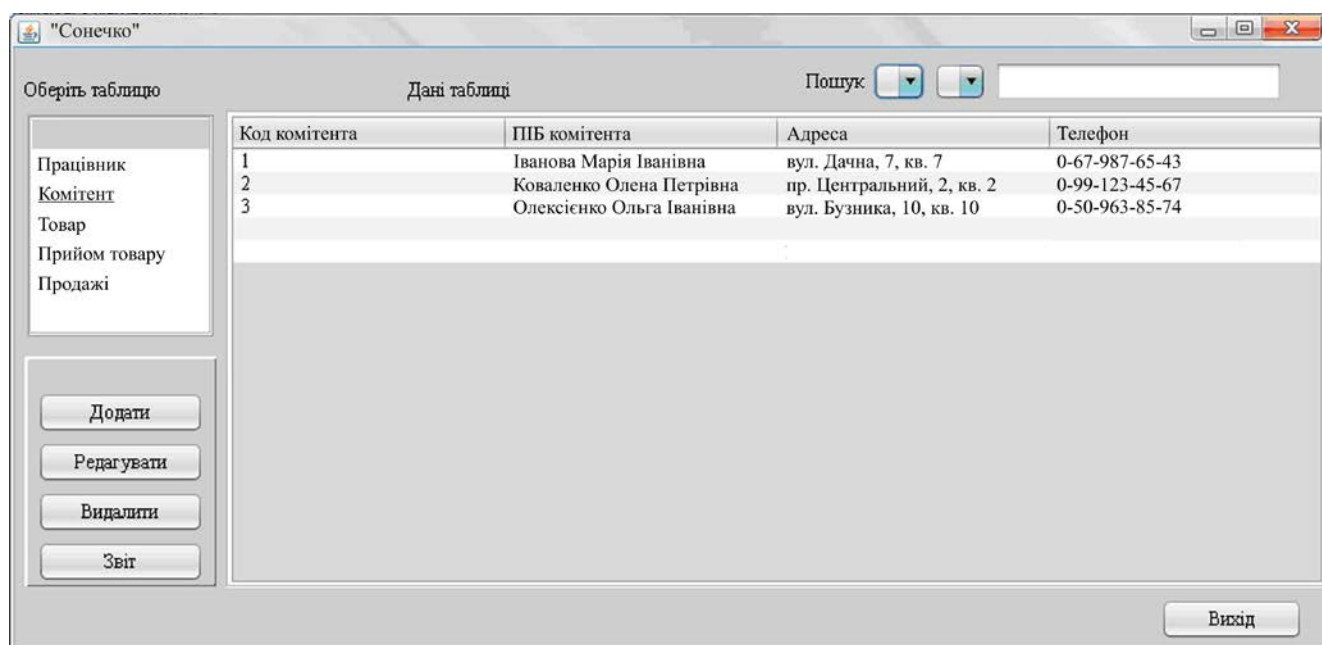


Рисунок 3.1 – Зовнішній вигляд головного вікна програма для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

Проведено огляд літературних джерел, інтернет-джерел для аналізу роботи та методів можливої автоматизації роботи з комітентами у дитячому комісійному

магазині "Сонечко". Проведено аналіз існуючих програмних рішень для автоматизації роботи.

Проведено аналіз вимог до програмного забезпечення, розроблено архітектуру та проєкт програмного забезпечення для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко", виконані реалізація та тестування розробленого програмного забезпечення.

В рамках аналізу вимог до програми було побудовано модель варіантів використання та були розроблені специфікації варіантів використання.

В рамках ескізного проєкту засобами мови моделювання UML побудовані сценарії варіантів використання, розроблена концептуальна модель даних та ескізи інтерфейсу програми.

В рамках технічного проєкту побудована діаграма класів, розроблені специфікації класів, побудовані діаграми послідовності, розроблена логічної моделі бази даних.

В рамках робочого проєкту виконано вибір і обґрунтування інструментів розробки та системи управління базою даних, побудована фізична модель бази даних, виконано кодування, тестування та випробування програми.

Також була розроблена уся необхідна програмна документація: технічне завдання (Додаток А), опис програми (Додаток В), інструкція користувача (Додаток Г), програма та методика випробування програмного забезпечення (Додаток Д). Наведено роздрук фрагментів коду програмного забезпечення (Додаток Б).

4 ОХОРОНА ПРАЦІ

4.1 Загальні питання організації охорони праці

Охорона праці становить багатокomпонентну систему правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів, які спрямовані на збереження життя, зміцнення здоров'я та підтримку працездатності працівників у процесі здійснення ними трудової діяльності. Ця система є невід'ємною складовою функціонування будь-якого виробничого чи невиробничого середовища, оскільки гарантує дотримання принципів безпеки, гігієни праці, а також реалізацію законного права людини на належні умови праці [19].

На сучасному етапі соціально-економічного розвитку України проблеми, пов'язані з ефективністю функціонування системи охорони праці, набувають особливої актуальності. Незважаючи на наявність достатньої кількості нормативно-правових актів у сфері охорони праці, практична реалізація багатьох із них залишається на недостатньому рівні. Це зумовлено як загальною нестабільністю соціально-економічної ситуації, так і низьким рівнем контролю за дотриманням трудового законодавства з боку державних інституцій. Однією з ключових проблем є відсутність офіційного працевлаштування значної частини населення, що унеможлиблює повноцінну реалізацію прав працівників на безпечні умови праці та відповідне соціальне забезпечення у разі професійних ризиків або нещасних випадків на виробництві.

Система охорони праці охоплює низку взаємопов'язаних компонентів, серед яких особливе місце займають: дотримання правових норм трудового законодавства, впровадження сучасних засобів техніки безпеки, підтримання належного санітарно-гігієнічного стану на робочих місцях, протипожежна безпека, електробезпека, а також постійний медичний контроль за станом здоров'я працівників. Комплексне поєднання цих заходів сприяє не лише

зниженню виробничого травматизму, але й підвищенню загального рівня трудової культури та ефективності виробничого процесу.

Основною метою охорони праці є забезпечення таких умов на підприємстві чи в організації, які б відповідали встановленим державним стандартам безпеки, не створювали ризиків для життя та здоров'я персоналу, а також сприяли підвищенню рівня мотивації працівників до сумлінного виконання своїх функціональних обов'язків. У цьому контексті охорона праці виступає не лише як форма захисту працівника, але й як стратегічний інструмент забезпечення стійкості й конкурентоспроможності підприємств у сучасних умовах.

4.2 Вплив виробничого середовища при роботі з персональним комп'ютером

Виробниче середовище – це сукупність фізичних, хімічних, біологічних та соціальних факторів, які безпосередньо впливають на працівника в процесі його трудової діяльності. Для офісних працівників, які щоденно працюють з персональним комп'ютером, найбільш актуальними є мікроклімат, освітлення, шум, електромагнітне поле, якість повітря, а також іонізація повітря.

Мікроклімат

Мікроклімат робочого середовища – один із головних факторів, що впливають на самопочуття і працездатність працівників. Він визначається:

- температурою повітря,
- відносною вологістю,
- швидкістю руху повітря,
- тепловим випромінюванням від приладів та обладнання.

Нормативи:

- Температура повітря: 18–24 °C (в залежності від сезону);
- Відносна вологість: 40–60%;

- Швидкість руху повітря: 0,1–0,2 м/с.

Порушення цих параметрів призводить до:

- перегрівання або переохолодження організму;
- зниження працездатності;
- загострення хронічних захворювань;
- виникнення простудних хвороб (при протягах або різких перепадах температури).

Заходи: використання кондиціонерів, зволожувачів повітря, встановлення систем припливно-витяжної вентиляції з регулюванням параметрів мікроклімату.

Освітлення

Рівень і якість освітлення безпосередньо впливають на зоровий апарат працівника. Недостатнє або надмірне освітлення викликає зорове перенапруження, головний біль, зниження зору.

Нормативи:

- Освітленість робочої поверхні: не менше 500 лк для офісних приміщень;
- Індекс передачі кольору (Ra): ≥ 80 ;
- Співвідношення природного і штучного світла: перевага надається природному освітленню зліва.

Проблеми:

- Наявність відблисків на екрані;
- Контраст між екраном і фоном;
- Часте мерехтіння ламп (пульсації).

Заходи: використання антивідблискових фільтрів, локальних джерел світла (настільних ламп), люмінесцентних або LED-світильників з плавним регулюванням яскравості.

Шум

Навіть у офісному середовищі шум може бути значним – через роботу систем охолодження, офісної техніки, вентиляторів тощо. Шум впливає на психоемоційний стан, викликає дратівливість, втомлюваність, зниження концентрації.

Нормативи:

- Рівень шуму в офісі не повинен перевищувати 50 дБА.

Джерела шуму:

- вентилятори системного блоку,
- принтери,
- кондиціонери,
- розмови співробітників.

Заходи: застосування тихої техніки, розміщення шумових пристроїв у шумоізованих нішах, використання звукопоглинаючих матеріалів у дизайні приміщення.

Іонізація повітря

Під час тривалої роботи комп'ютерів змінюється іонний склад повітря:

- зменшується концентрація негативних іонів (корисних),
- зростає концентрація позитивних іонів (шкідливих),

що викликає:

- головний біль,
- погіршення самопочуття,
- зниження імунітету,
- підвищену стомлюваність.

Нормативне співвідношення іонів:

- негативні іони: не менше 400 іонів/см³;
- позитивні – не більше 2000 іонів/см³.

Заходи: використання іонізаторів повітря, регулярне провітрювання, кімнатні рослини (наприклад, фікус, алое, хлорофітум), які природним чином покращують склад повітря.

Електромагнітне поле та випромінювання

Робочі місця з персональними комп'ютерами піддаються впливу низькочастотних електромагнітних полів, особливо у зоні монітора, кабелів, розеток. Хоч сучасні ЖК-монітори мають мінімальний рівень випромінювання, тривалий вплив все одно вимагає контролю.

Нормативи:

- Електростатичне поле: не більше 15 кВ/м;
- Магнітне поле: не більше 0,3 мТл (мілітесла).

Наслідки впливу:

- підвищена стомлюваність,
- безсоння,
- нервова збудливість.

Заходи: застосування екранів, заземлення техніки, правильне розташування обладнання, дотримання відстані від монітора до очей – не менше 60 см.

Забруднення повітря (хімічні фактори)

При роботі офісного обладнання (принтери, сканери, факси) виділяються дрібнодисперсні частинки, леткі органічні сполуки, формальдегід, озон.

Джерела:

- матеріали корпусів ПК,
- офісні меблі (ЛДСП, МДФ),
- чорнила та тонери.

Наслідки:

- алергії,
- подразнення слизових оболонок,
- хронічні респіраторні захворювання.

Заходи: встановлення витяжної вентиляції, часте провітрювання, використання матеріалів з сертифікатом екологічної безпеки.

Висновок

Виробниче середовище в офісі або у приміщенні для роботи з персональним комп'ютером має бути ретельно контрольоване. Забезпечення нормативних мікрокліматичних умов, достатньої вентиляції, ергономічного освітлення, низького рівня шуму та оптимального складу повітря є ключовими умовами збереження здоров'я працівника та підвищення його продуктивності.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було розв'язано практичну задачу інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов – розроблена програма для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко".

Для досягнення поставленої мети були вирішені наступні задачі:

- проведено аналіз практичної задачі інженерії програмного забезпечення, яка вирішується у кваліфікаційній роботі;
- зроблено висновок про доцільність розробки власної програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко";
- сформульовані вимоги до програмного забезпечення та розроблено технічне завдання;
- розроблений проєкт програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко", який включає ескізний, технічний та робочий проєкти цього програмного забезпечення;
- проведено тестування та випробування розробленої програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко";
- розроблено усю необхідну документацію до програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко".

В результаті була розроблена програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко", яка дозволила спростити та автоматизувати роботу з комітентами, зокрема полегшити ведення обліку розрахунків та формування відповідної звітності, зменшити кількість помилок.

Також було виконано розділ з охорони праці.

Мета кваліфікаційної роботи досягнута, усі завдання виконані в повному обсязі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ефективність автоматизації бізнес-процесів: від теорії до практики. URL: <https://ternopolis.te.ua/2024/02/efektivnist-avtomatizaciyi-biznes-procesiv-vid-teoriyi-do-praktiki/> (дата звернення: 03.03.2025).
2. Автоматизація роздрібних магазинів, butikiv і аптек з використанням продуктів сімейства МініСофт Комерція. URL: <https://minisoft.ua/article/avtomatizaciya-rozdribnikh-magaziniv-butikiv-i-aptek-z-vikoristannyam-produktiv-simeystva> (дата звернення: 03.03.2025).
3. Сімейство програм МініСофт Комерція. URL: <https://minisoft.ua/po/simeystvo-program-minisoft-komerciya> (дата звернення: 03.03.2025).
4. Автоматизація магазину. URL: <https://skynum.ua/solutions/shop> (дата звернення: 03.03.2025).
5. Система для автоматизації та обліку вашої торгової мережі. URL: <https://paykitpos.com/avtomatyzacziyi-torgovoyi-merezhi> (дата звернення: 03.03.2025).
6. Автоматизація магазину роздрібної торгівлі з AinurPOS. URL: <https://ainurpos.com.ua/avtomatyzacziya-magazynu-rozdribnoyi-torgivli-z-ainurpos/> (дата звернення: 03.03.2025).
7. Роздрібна торгівля. URL: <https://torgsoft.ua/soft/retail/> (дата звернення: 03.03.2025).
8. Проектування та розробка корпоративних інформаційних систем. Концептуальні моделі та схеми даних. URL: http://citforum.ck.ua/cfin/prcorpsys/infersistpr_09.shtml (дата звернення: 05.04.2025).
9. Архітектура програмного забезпечення: все що треба знати. URL: <https://wezom.com.ua/ua/blog/arhitektura-programmnogo-obespecheniya> (дата звернення: 05.04.2025).

10. Діаграма діяльності в UML: символ, компоненти та приклад. URL: <https://www.guru99.com/uk/uml-activity-diagram.html> (дата звернення: 05.04.2024).

11. Модель «сутність-зв'язок». URL: https://virt.ldubgd.edu.ua/pluginfile.php/38821/mod_resource/content/1/ (дата звернення: 20.04.2025)

12. Інтерфейс користувача. URL: <http://sven.ua/ua/service/glossary/ukr/11/> (дата звернення: 20.04.2025).

13. Діаграма класів. URL: <http://moodle.ipo.kpi.ua/moodle/mod/resource/view.php?id=28088> (дата звернення: 20.04.2025).

14. Діаграма взаємодії. URL: <http://moodle.ipo.kpi.ua/moodle/mod/resource/view.php?r=16538> (дата звернення: 20.04.2025).

15. Докладніше про технологію Java. URL: <https://www.java.com/about/> (дата звернення: 11.05.2025).

16. The GNU General Public License. URL: <http://www.gnu.org/licenses/licenses.en.html> (дата звернення: 11.05.2025).

17. MySQL. The world's most popular open source database. URL: <https://www.mysql.com/> (дата звернення: 11.05.2025).

18. IntelliJ IDEA Ultimate. The Leading Java and Kotlin IDE. URL: <https://www.jetbrains.com/idea/download/?section=windows> (дата звернення: 11.05.2025).

19. Про охорону праці: Закон України від 21.11.2002 р. № 229-IV. Дата оновлення: 04.02.2021. URL: <https://zakon.rada.gov.ua/laws/main/2694-12#Text>. (дата звернення: 17.03.2025).

ДОДАТОК А

Технічне завдання на розробку програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

Вступ

Назва програмного забезпечення – "Програма для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко", далі – "програмне забезпечення".

Галузь застосування: програмний продукт призначений для застосування у дитячому комісійному магазині "Сонечко".

1. Підстави для розробки

Підставою для розробки програмного забезпечення є завдання на кваліфікаційну роботу на здобуття ступеня вищої освіти «бакалавр», виданого кафедрою ПЗАС НУК імені адмірала Макарова.

2. Призначення розробки

2.1 Експлуатаційне призначення

Експлуатаційним призначенням програмного забезпечення є спрощення та автоматизація роботи з комітентами у дитячому комісійному магазині "Сонечко".

2.2 Функціональне призначення

Функціональним призначенням програмного забезпечення є автоматизація процесів:

- реєстрації та авторизації користувачів;
- зберігання та використання інформації про комітентів;
- зберігання та використання інформації про товар;
- зберігання та використання інформації з продажу товару;
- складання звітності.

3. Вимоги до програмного продукту

3.1 Вимоги до функціональних характеристик

3.1.1 Вимоги до складу виконуваних функцій

Програма повинна виконувати наступні функції.

Авторизація користувача

Авторизація користувача в програмному забезпеченні для надання функціональних можливостей. Обов'язковими полями є: логін та пароль.

Перегляд даних

Для перегляду даних потрібно обрати таблицю.

Пошук необхідної інформації

Функція пошуку необхідної інформації відбувається за певними параметрами: необхідно обрати таблицю та вести атрибути для пошуку.

Додавання нового запису

Для додавання нового запису потрібно обрати таблицю.

Редагування запису

Для редагування запису потрібно обрати таблицю та відповідний запис.

Видалення запису

Для видалення запису потрібно обрати таблицю та відповідний запис.

Формування документу про прийом товару від комітента

Формування накладної з переліком товарів, переданих на реалізацію до магазину.

Оформлення продажу товару

Фіксування факту продажу конкретного товару.

Формування звітів

Формування звітів повинно відбуватися автоматично після обрання необхідних відповідних параметрів.

Представлення даних

Данні представляються у вигляді наступних таблиць у БД: працівник,

комітент, товар, продажі. Атрибути таблиць та їх опис наведено у таблицях А.1 – А.5.

Таблиця А.1 – Опис атрибутів таблиці БД «Працівник»

Атрибут	Тип даних
Код працівника	Число
ПІБ працівника	Текст
Адреса	Текст
Телефон	Текст
Дата прийняття на роботу	Дата
Дата звільнення з роботи	Дата

Таблиця А.2 – Опис атрибутів таблиці БД «Комітент»

Атрибут	Тип даних
Код комітента	Число
ПІБ комітента	Текст
Адреса	Текст
Телефон	Текст

Таблиця А.3 – Опис атрибутів таблиці БД «Товар»

Атрибут	Тип даних
Код товару	Число
Тип товару	Текст
Найменування товару	Текст

Таблиця А.4 – Опис атрибутів таблиці БД «Прийом товару»

Атрибут	Тип даних
Код прийому	Число
Код комітента	Число
Код товару	Число
Кількість	Число
Ціна продажу	Число

Таблиця А.5 – Опис атрибутів таблиці БД «Продажі»

Атрибут	Тип даних
Код продажі	Число
Код працівника	Число
Код комітента	Число
Код прийому	Число
Кількість	Число
Дата продажу	Дата

3.1.2 Вимоги до організаційних вхідних та вихідних даних

Дані зберігаються в файлі БД з довільним доступом типу .myd.

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 та принтер).

3.2 Вимоги до надійності

Програмний продукт повинен нормально функціонувати при безперебійній роботі комп'ютера. При виникненні збоїв в роботі, відновлення нормальної роботи повинне проводитися після перезавантаження програми.

Для забезпечення надійності інформації повинна використовуватися СКБД, що забезпечує цілісність транзакцій і несе відповідальність за цілісність інформації.

Система повинна продовжувати коректне функціонувати при втраті частини інформації. У випадку неможливого продовження коректної роботи має бути повідомлення про це.

У разі введення користувачем некоректної інформації система повинна повідомити про помилку і надати можливість виправити її.

3.3 Вимоги до умов експлуатації

Кліматичні умови експлуатації, при яких повинні забезпечуватися задані характеристики, повинні задовольняти вимогам, пропонованим до технічних засобів у частині умов їхньої експлуатації.

3.4 Вимоги до складу та параметрів технічних засобів

Система повинна задовольняти наступним вимогам в мінімальній комплектації:

– CPU: Intel(R) i5 2,16 МГц або аналогічний за параметрами AMD;

- RAM: 4 GB;
- HDD: 10 GB вільного місця;
- пристрої вводу-виводу: монітор з екраном роздільної здатності не менше ніж 1600x900 пікселів, клавіатура, маніпулятор типу «миша».

3.5 Вимоги до інформаційної та програмної сумісності

Для повноцінного функціонування системи необхідно використовувати операційну систему Windows версії не нижче 7. Необхідна наявність встановленого Standard Edition Development Kit не нижче 17.

3.6 Вимоги до маркування та пакування

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

- виконуваний файл `sunshine.jar`;
- файл з описом `ReadMe.txt`;
- файл, який містить структуру БД `sunshine.sql`.

3.7 Вимоги до транспортування та зберігання

Вимоги до транспортування не висовуються у зв'язку з відсутністю фізичних носіїв.

4 Вимоги до програмної документації

Програмна документація повинна містити:

- технічне завдання;
- код програмних модулів;
- опис програми;
- керівництво користувача;
- програму та методику випробування програмного забезпечення.

5 Техніко-економічні показники

Техніко-економічні показники не розраховуються у зв'язку з тим, що програмного забезпечення розробляється в рамках кваліфікаційної роботи.

6 Стадії та етапи розробки

Стадії та етапи розробки представлені в таблиці А.6.

Таблиця А.6 – Стадії та етапи розробки програмного забезпечення

№ п/п	Назва етапів роботи	Терміни виконання
1.	Аналіз практичної задачі інженерії ПЗ	05.02.2025 – 12.02.2025
2.	Аналіз вимог до ПЗ	12.02.2025 – 20.02.2025
3.	Розробка архітектури ПЗ	20.02.2025 – 03.03.2025
4.	Розробка проекту ПЗ	03.03.2025 – 15.04.2025
5.	Реалізація та тестування ПЗ	15.04.2025 – 12.05.2025

7 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 проводяться відповідно до програми і методики випробувань і документують за допомогою протоколу проведення випробувань. У разі знаходження помилок під час прийому програмного забезпечення складається акт про знайдені помилки, який підписуються представниками замовника і розробника і затверджуються керівником організації – замовника та організації – розробника. Розробник повинен на протязі не більше ніж 2 тижнів виправити зазначені помилки і оповістити замовника про повторне проведення перевірки.

Додаток Б Код програми для автоматизації роботи з комітентами дитячого
комісійного магазину "Сонечко"

```
-- Table: Employee (співробітник)
CREATE TABLE Employee (
    employee_id INT PRIMARY KEY,
    full_name VARCHAR(255) NOT NULL,
    address VARCHAR(255),
    phone VARCHAR(20),
    hire_date DATE,
    dismissal_date DATE
);

-- Table: Consignor (комітент)
CREATE TABLE Consignor (
    consignor_id INT PRIMARY KEY,
    full_name VARCHAR(255) NOT NULL,
    address VARCHAR(255),
    phone VARCHAR(20)
);

-- Table: Product (товар)
CREATE TABLE Product (
    product_id INT PRIMARY KEY,
    product_type VARCHAR(100),
    product_name VARCHAR(255)
);

-- Table: ProductReceipt (Прийом товару)
CREATE TABLE ProductReceipt (
    receipt_id INT PRIMARY KEY,
    consignor_id INT,
    product_id INT,
    quantity INT NOT NULL,
    selling_price DECIMAL(10, 2),

    CONSTRAINT fk_receipt_consignor FOREIGN KEY (consignor_id)
REFERENCES Consignor(consignor_id),
    CONSTRAINT fk_receipt_product FOREIGN KEY (product_id)
REFERENCES Product(product_id)
);

-- Table: Sale (продаж)
CREATE TABLE Sale (
    sale_id INT PRIMARY KEY,
    employee_id INT,
    consignor_id INT,
    receipt_id INT,
    quantity INT NOT NULL,
```

```

        sale_date DATE,

        CONSTRAINT fk_sale_employee FOREIGN KEY (employee_id) REFERENCES
Employee(employee_id),
        CONSTRAINT fk_sale_consignor FOREIGN KEY (consignor_id)
REFERENCES Consignor(consignor_id),
        CONSTRAINT fk_sale_receipt FOREIGN KEY (receipt_id) REFERENCES
ProductReceipt(receipt_id)
);

```

```

package com.boss;
import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Locale;
import java.util.Map;
import java.util.Vector;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.table.DefaultTableModel;
import net.sf.jasperreports.engine.JRException;
import net.sf.jasperreports.engine.JasperExportManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.view.JasperViewer;
public class DBHelper {
    public static Connection connection = null;
    final private static Vector<Vector> tableData = new Vector<>();
    final private static Vector columnNames = new Vector();
    final private static Vector row = new Vector();
    private static final DefaultTableModel tableModel = new
DefaultTableModel();
    public static ResultSet resultSet = null;
    final public static ArrayList id = new ArrayList();
    public static Statement statement = null;
    public static String CURRENT_TABLE;
    static public Connection openConnection(String db) {
        if (connection == null) {
            try {
                Class.forName("org.sqlite.JDBC");
                System.out.println(db);
                connection = (Connection)
DriverManager.getConnection("jdbc:sqlite:db/" + db);
                if (connection != null) {
                    System.out.println("Connected");
                } else {

```

```

        System.out.print("Connection failed");
    }
    } catch (ClassNotFoundException | SQLException ex) {
        System.err.println(ex.getCause());
    }
}
return connection;
}
public static void closeConnection() {
    try {
        if (statement != null && !statement.isClosed()) {
            statement.close();
        }
        getConnection().close();
        System.out.println("Connection closed.");
    } catch (SQLException ex) {
        System.err.println(ex.getCause());
    }
}
public static DefaultTableModel executeSelectQuery(String... q)
{
    tableData.clear();
    columnNames.clear();
    CURRENT_TABLE = q[0];
    String query = "Select * from " + CURRENT_TABLE + " " +
(q.length == 1 ? "" : q[1]);
    System.out.println(query);
    try {
        resultSet = getStatement().executeQuery(query);
        initColumnNames(resultSet.getMetaData());
        while (resultSet.next()) {
            row.clear();
            for (int i = 0; i < columnNames.size(); i++) {
                row.add(resultSet.getObject(i + 1));
            }
            tableData.add((Vector) row.clone());
        }
        tableModel.setDataVector(tableData, columnNames);
        return tableModel;
    } catch (SQLException e) {
        e.printStackTrace();
        return new DefaultTableModel();
    }
}
public static void executeInsertQuery(Vector colNames, Vector
rowValues) {
    StringBuilder query = new StringBuilder("INSERT INTO ");
    StringBuilder queryValues = new StringBuilder(" VALUES ('");
    query.append(CURRENT_TABLE).append(" (");
    for (int i = 0; i < rowValues.size(); i++) {
        if (i != rowValues.size() - 1) {
            query.append(colNames.get(i)).append(",");
            queryValues.append(rowValues.get(i)).append(",");
        }
    }
    query.append(")");
    queryValues.append(")");
    executeQuery(query.toString(), queryValues.toString());
}

```

```

        } else {
            query.append(colNames.get(i)).append(" ");
            queryValues.append(rowValues.get(i)).append("'");
        }
    }
    query.append(queryValues);
    System.out.println(query);
    try {
        getStatement().execute(query.toString());
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

public static void executeDeleteQuery(int indexToDelete) {
    try {
        getStatement().execute("Delete from " + CURRENT_TABLE +
" Where ID = " + indexToDelete);
        System.out.println("Index " + indexToDelete + " from " +
CURRENT_TABLE + " was deleted;");
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

public static void executeUpdateQuery(Vector colNames, Vector
rowValues, int ID) {
    StringBuilder query = new StringBuilder("UPDATE ");
    query.append(CURRENT_TABLE).append(" SET ");
    for (int i = 0; i < rowValues.size(); i++) {
        query.append(colNames.get(i)).append("=").append(rowValues.get(i)).
append("' ");
        if (i != rowValues.size() - 1) {
            query.append(", ");
        } else {
            query.append(" WHERE ID = ").append(ID);
        }
    }
    System.out.println(query);
    try {
        getStatement().execute(query.toString());
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

private static void initColumnNames(ResultSetMetaData metaData)
throws SQLException {
    for (int i = 0; i < metaData.getColumnCount(); i++) {
        columnNames.add(metaData.getColumnName(i + 1));
    }
}

private static Connection getConnection() throws SQLException {
    if (connection == null || connection.isClosed()) {
        System.err.print("Connection does not exists.");
    }
}

```

```

        return connection;
    }
    private static Statement getStatement() throws SQLException {
        if ((statement == null || statement.isClosed()) &&
(connection != null)) {
            statement = getConnection().createStatement();
        }
        return statement;
    }
    public static Vector getColumnNames() {
        return columnNames;
    }
    /**
     * Returns column names without ID column
     *
     * @return
     */
    public static Vector getExtraColumnNames() {
        Vector extraColumnNames = (Vector) columnNames.clone();
        extraColumnNames.remove(0);
        return extraColumnNames;
    }
    public static Vector<String> getTables() {
        Vector<String> tableNames = new Vector<>();
        try {
            DatabaseMetaData metaData = connection.getMetaData();
            resultSet = metaData.getTables("", "", "%", null);
            while (resultSet.next()) {
                if (!resultSet.getString(3).startsWith("sqlite_")) {
                    tableNames.add(resultSet.getString(3));
                }
            }
        } catch (SQLException ex) {
            Logger.getLogger(DBHelper.class.getName()).log(Level.SEVERE, null,
ex);
        }
        return tableNames;
    }

```

```

package com.boss.view;
import javax.swing.JOptionPane;
public class Autorization extends javax.swing.JFrame {
    public Autorization() {
        initComponents();
    }
    @SuppressWarnings("unchecked")
    private void initComponents() {

```

```

LoginjLabel1 = new javax.swing.JLabel();
PasswjLabel1 = new javax.swing.JLabel();
PasswordjTextField1 = new javax.swing.JTextField();
EnterjButton1 = new javax.swing.JButton();
jLabel1 = new javax.swing.JLabel();
ExitjButton1 = new javax.swing.JButton();
LoginjTextField1 = new javax.swing.JTextField();
LogojLabel2 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Сонечко");
setBackground(new java.awt.Color(255, 255, 255));
setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
LoginjLabel1.setFont(new java.awt.Font("Times New Roman", 0, 14));
// NOI18N
LoginjLabel1.setText("Логін");
PasswjLabel1.setFont(new java.awt.Font("Times New Roman", 0, 14));
// NOI18N
PasswjLabel1.setText("Пароль");
PasswordjTextField1.setFont(new java.awt.Font("Times New Roman", 0,
14)); // NOI18N
PasswordjTextField1.setText("");
PasswordjTextField1.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        PasswordjTextField1ActionPerformed(evt);
    }
});
EnterjButton1.setFont(new java.awt.Font("Times New Roman", 0, 14));
// NOI18N
EnterjButton1.setText("Вхід");
EnterjButton1.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
EnterjButton1.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        EnterjButton1ActionPerformed(evt);
    }
});
jLabel1.setFont(new java.awt.Font("Monospaced", 1, 24)); // NOI18N

```

```

jLabel1.setText("Сонечко");
ExitjButton1.setFont(new java.awt.Font("Times New Roman", 0, 14));
// NOI18N
ExitjButton1.setText("Вихід");
ExitjButton1.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
ExitjButton1.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ExitjButton1ActionPerformed(evt);
    }
});
LoginjTextField1.setFont(new java.awt.Font("Times New Roman", 0,
14)); // NOI18N
LoginjTextField1.setText("");
LoginjTextField1.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        LoginjTextField1ActionPerformed(evt);
    }
});
LogoJLabel2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/ic1.PNG"))); // NOI18N
javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(jLabel1))
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createSequentialGroup()
                        .addGap(76, 76, 76)

```

```

javax.swing.GroupLayout.PREFERRED_SIZE, 158,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(EnterjButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 158,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addGroup(layout.createSequentialGroup())

        .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(LoginjLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(PasswjLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(37, 37, 37)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup())

            .addComponent(LogojLabel2)

            .addGap(0, 0, Short.MAX_VALUE))

            .addComponent(LoginjTextField1)

            .addComponent>PasswordjTextField1))))

        .addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup())

            .addContainerGap()

            .addComponent(jLabel1)

            .addGap(34, 34, 34)

            .addComponent(LogojLabel2)

            .addGap(31, 31, 31)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(LoginjLabel1)

        .addComponent(LoginjTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,

```

```

javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(PasswjLabel1)

        .addComponent(PasswordjTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(18, 18, 18)

        .addComponent(EnterjButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(ExitjButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

    );

    pack();

}

private void LoginjTextField1ActionPerformed(java.awt.event.ActionEvent
evt) {

}

private void
PasswordjTextField1ActionPerformed(java.awt.event.ActionEvent evt) {

}

private void EnterjButton1ActionPerformed(java.awt.event.ActionEvent
evt) {

    String login = LoginjTextField1.getText();
    String password = PasswordjTextField1.getText();
    if (login == null || login.isEmpty()) {
        JOptionPane.showMessageDialog(rootPane, "Введіть логін");
    } else if (password == null || password.isEmpty()) {
        JOptionPane.showMessageDialog(rootPane, "Введіть пароль");
    } else if ("".equals(login.trim()) && "".equals(password.trim())) {
        dispose();
        new View();
    }
}

```

```

        } else {
            JOptionPane.showMessageDialog(rootPane, "Логін або пароль не
вірний");
        }
    }

    private void ExitjButton1ActionPerformed(java.awt.event.ActionEvent
evt) {
        dispose();
    }
    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException | InstantiationException |
IllegalAccessException | javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(Autorization.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
        }
        java.awt.EventQueue.invokeLater(new Runnable() {
            @Override
            public void run() {
                new Autorization().setVisible(true);
            }
        });
    }

    private javax.swing.JButton EnterjButton1;
    private javax.swing.JButton ExitjButton1;
    private javax.swing.JLabel LoginjLabel1;
    private javax.swing.JTextField LoginjTextField1;
    private javax.swing.JLabel LogojLabel2;

```

```
private javax.swing.JLabel PasswjLabel1;  
private javax.swing.JTextField PasswordjTextField1;  
private javax.swing.JLabel jLabel1;  
}
```

Додаток В Опис програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

1 Загальні відомості

Розроблена програма для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" представляє собою програму для співробітників дитячого комісійного магазину "Сонечко".

Програма призначена для спрощення та автоматизації роботи з комітентами, зокрема полегшення ведення обліку розрахунків та формування відповідної звітності, зменшення кількості помилок.

Для розробки програми були обрані наступні засоби:

- в якості мови програмування була обрана Java;
- в якості СКБД була обрана MySQL;
- в якості середовища розробки було обрано середовище JetBrains IntelliJ IDEA.

2 Функціональне призначення

Програма виконує наступні основні функції:

- реєстрації та авторизації користувачів;
- зберігання та використання інформації про комітентів;
- зберігання та використання інформації про товар;
- зберігання та використання інформації з продажу товару;
- складання звітності.

3 Опис логічної структури

Програма розроблена на модульній основі, тобто кожний компонент

(модуль) програми відповідає за конкретну дію. Дані, необхідні для роботи програми, зберігаються у нормалізованих таблицях бази даних.

4 Технічні та програмні засоби

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

- CPU: Intel(R) i5 2,16 МГц або аналогічний за параметрами AMD;
- RAM: 4 GB;
- HDD: 10 GB вільного місця;
- пристрої вводу-виводу: монітор з екраном роздільної здатності не менше ніж 1600x900 пікселів, клавіатура, маніпулятор типу «миша».

Для повноцінного функціонування системи необхідно використовувати ОС Windows версії не нижче 7.

Необхідна наявність встановленого Standard Edition Development Kit не нижче 17.

5 Виклик та завантаження

Програма для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" встановлюється безпосередньо на комп'ютер користувача. Програма відкривається або за допомогою ярлика, який знаходиться на робочому столі, або за допомогою вибору відповідного пункту меню «Пуск» – «Програми». Програма має один режим використання.

Після запуску програми з'являється вікно авторизації користувача, де пропонується ввести логін та пароль для входу в систему. Якщо дані користувача введені вірно, відкривається головна форма програми.

6 Вхідні та вихідні дані

Дані зберігаються в файлі БД з довільним доступом типу .myd.

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 та принтер).

Додаток Г Інструкція користувача програми для автоматизації роботи з
комітентами дитячого комісійного магазину "Сонечко"

Програма для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко" встановлюється безпосередньо на комп'ютер користувача. Програма відкривається або за допомогою ярлика, який знаходиться на робочому столі, або за допомогою вибору відповідного пункту меню «Пуск» – «Програми». Програма має один режим використання.

Після запуску програми з'являється вікно авторизації користувача, де пропонується ввести логін та пароль для входу в систему. Вікно авторизації наведено на рис. Г.1.

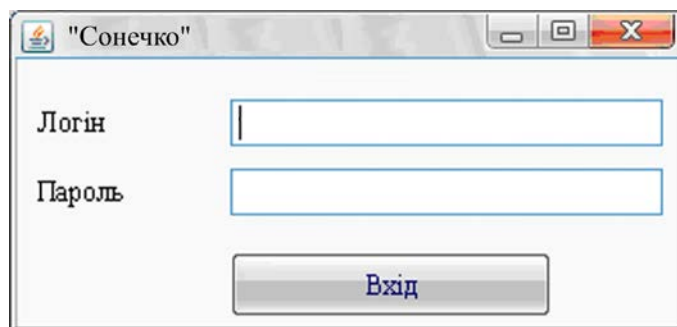


Рисунок Г.1 – Вікно авторизації користувача програми для автоматизації роботи з
комітентами дитячого комісійного магазину "Сонечко"

Якщо було введено не вірні дані користувача, з'являється діалогове вікно з попередженням, далі програма надає змогу для повторного введення даних. Якщо дані користувача введені вірно, відкривається головна форма програми, наведена на рис. Г.2.

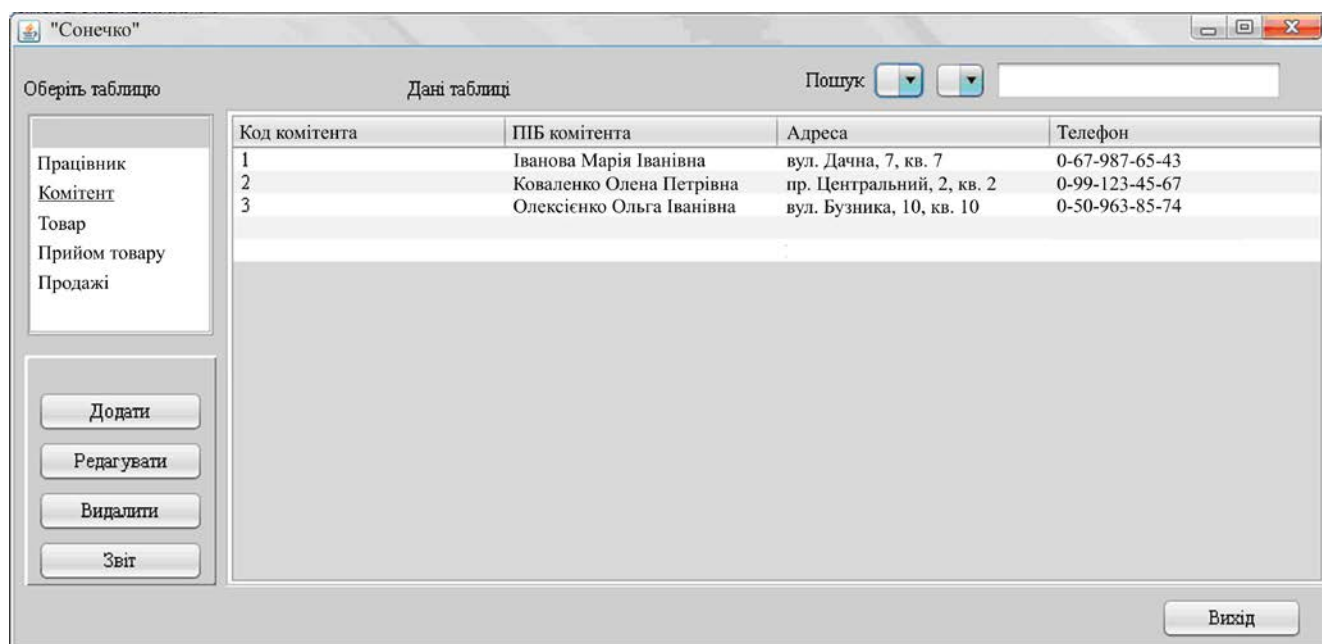


Рисунок Г.2 – Головна форма програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

Для роботи із записами таблиці необхідно спочатку обрати таблицю зі списку доступних таблиць, що розташований в лівій частині головної форми програми. Після цього дані з обраної таблиці з'являться на панелі «Дані таблиці», що наведено на рис. Г.3.

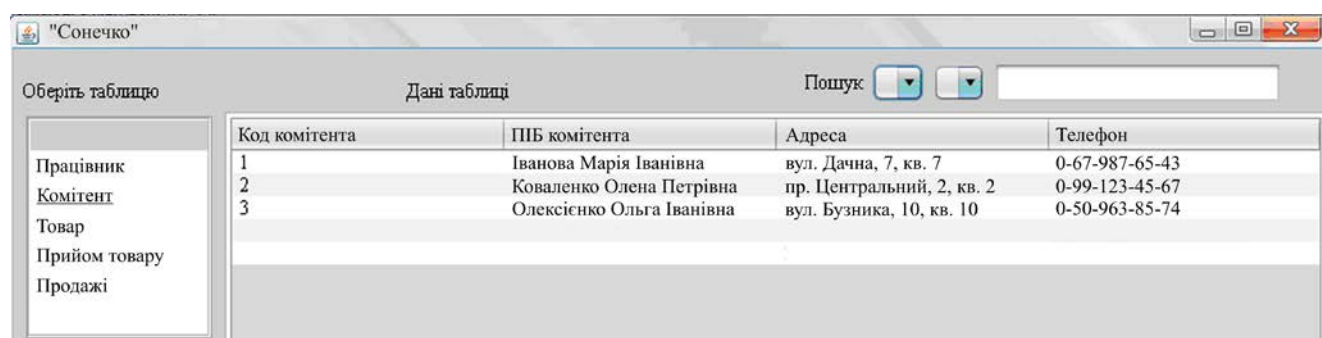


Рисунок Г.3 – Обрана таблиця та панель «Дані таблиці»

Для додавання нового запису у таблицю необхідно:

1. Обрати таблицю.
2. На головній формі натиснути кнопку «Додати» (див. рис. Г.2).

3. На формі для додавання нового запису у таблицю заповнити всі необхідні поля на натиснути кнопку «Зберегти». У разі помилки натиснути кнопку «Відмінити». Форму для додавання нового запису у таблицю наведено на рис. Г.4.

Код комітента	ПІБ комітента	Адреса	Телефон
1	Іванова Марія Іванівна	вул. Дачна, 7, кв. 7	0-67-987-65-43

Рисунок Г.4 – Форма для додавання нового запису

Для редагування запису в таблиці необхідно:

1. Обрати таблицю.
2. Обрати запис із таблиці.
3. На головній формі, натиснути кнопку «Редагувати» (див. рис. Г.2).
4. На формі для редагування запису з таблиці відредагувати всі необхідні поля на натиснути кнопку «Зберегти». У разі помилки натиснути кнопку «Відмінити».

Для видалення запису з таблиці необхідно:

1. Обрати таблицю.
2. Обрати запис із таблиці.
3. На головній формі натиснути кнопку «Видалити» (див. рис. Г.2).

Для друкування звіту необхідно:

1. Обрати таблицю «Комітент».
2. На головній формі натиснути кнопку «Звіт» (див. рис. Г.2).
3. У вікні «Друкувати звіт» обрати комітента з випадаючого списку та натиснути кнопку «Друкувати». Вікно «Друкувати звіт» з обраним комітентом наведено на рис. Г.5.

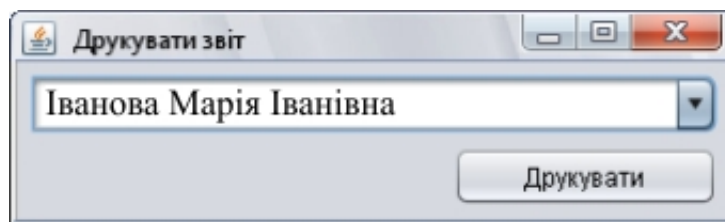


Рисунок Г.5 – Вікно «Друкувати звіт» з обраним комітентом

Додаток Д Програма та методика випробування програми для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко"

1. Об'єкт випробувань

1.1 Назва об'єкту

Повна назва об'єкта випробувань: Програма для автоматизації роботи з комітентами дитячого комісійного магазину "Сонечко".

Коротка назва: Програма.

1.2 Область застосування

Програма призначена для автоматизації роботи з комітентами у дитячому комісійному магазині "Сонечко", яка дозволить спростити та автоматизувати роботу з комітентами, зокрема полегшити ведення обліку розрахунків та формування відповідної звітності, зменшити кількість помилок.

Програма забезпечує:

- реєстрації та авторизації користувачів;
- зберігання та використання інформації про комітентів;
- зберігання та використання інформації про товар;
- зберігання та використання інформації з продажу товару;
- складання звітності.

2 Мета випробувань

Мета проведення випробувань – оцінка експлуатаційних характеристик програми, перевірка і підтвердження працездатності програми в умовах, максимально наближених до умов реальної експлуатації.

3 Вимоги до програми

Програма має реалізовувати функції, описані в технічному завданні, яке наведено в додатку А.

4 Вимоги до програмної документації

Для проведення тестування повинна бути надана наступна документація:

- технічне завдання на розробку програми;
- текст програми;
- опис програми;
- інструкція користувача;
- програма і методика випробувань програмного забезпечення.

5 Склад, порядок та методи випробувань

5.1 Перевірка програми на відповідність технічному завданню:

- перевірка роботи функції «Додавання запису»;
- перевірка роботи функції «Редагування запису»;
- перевірка роботи функції «Видалення запису»;
- перевірка роботи функції «Пошук інформації»;
- перевірка роботи функції «Облік прийому товару від комітента».

5.2 Перевірка програми на її програмно-апаратну сумісність: тестування на апаратних-програмних платформах різної конфігурації, але не нижче за мінімальні вимоги, наведені в технічному завданні; перевірка на наявність і усунення всіх помилок, зазначених у п. 5.1.

5.3 Візуальний перегляд програми: остаточне налагодження на предмет виявлення та усунення дрібних помилок.

6 Методи випробувань

Випробування проводиться за стратегією «чорного ящика». Через велику кількість функцій, які потрібно випробувати, представлено методи випробувань декількох функцій програми.

6.1. Перевірка програми на відповідність технічному завданню:

– перевірка роботи функції «Додавання запису»

На головному вікні програми обираємо таблицю та натискаємо кнопку «Додати». З'являється відповідне вікно, в якому вносимо необхідну інформацію. Після чого натискаємо кнопку «Зберегти».

– перевірка роботи функції «Пошук інформації»

На головному вікні програми обираємо таблицю, потім вводимо дані для пошуку, та натискаємо кнопку «Пошук». Повинні з'явитися результати пошуку.

– перевірка роботи функції «Облік прийому товару від комітента»

На головному вікні програми обираємо таблицю «Прийом товару» та натискаємо кнопку «Додати». З'являється відповідне вікно, у якому заповнюємо усі необхідні поля та натискаємо кнопку «Зберегти». У результаті повинні отримати сформований новий прийом товару від комітента.

6.2. Перевірка програми на її програмно-апаратну сумісність: виконуємо випробування на апаратно-програмних платформах різної конфігурації, але не нижче за мінімальні вимоги, наведені в технічному завданні; виконуємо перевірку на наявність і усунення всіх помилок, зазначених у п. 6.1.

6.3. Проводимо візуальний перегляд програми: виконуємо остаточне налагодження на предмет виявлення та усунення дрібних помилок.